SYSTEMS ANALYSIS OF PRODUCTION LINES(Dissertation_全文)

Yamashina, Hajime

SYSTEMS ANALYSIS
OF
PRODUCTION LINES

Hajime YAMASHINA

1973

SYSTEMS ANALYSIS OF PRODUCTION LINES

Hajime YAMASHINA

1973

INTRODUCTION

Control over the detailed operations of manufacturing processes has recently become a subject of basic research. Manufacturing processes may be described as production lines including continuous production or assembly operation. Each line may consist of a series of processing stages or facilities where either raw or partly finished materials are transformed or assembled into a relatively finished product. Three major problems in the design and operation of production lines in order to reduce costs and to improve productivity are :

(1) The sequencing of jobs into stations and the determination of the number of stations,

(2) The location of bunkers or storage spaces for in-process inventory,

(3) The size or capacity of these pulsating stores.

This kind of activity to make better decisions about these problems is called 'sequencing and in-process inventory control of production lines.'

This dissertation presents systems engineering approaches to these problems in order to aid management in decision-making for the development of highly efficient production lines. Such studies are essential for the advent of truly automated production control systems.

PART I of the dissertation is primarily concerned with the first problem, and PART II gives an exposition of the second and the third problems, mentioned above.

CONTENTS

page

page

page

page

PART I

SEQUENCING OF PRODUCTION LINES

INTRODUCTION

Sequencing problems are very commonly encountered. They are encountered whenever there is a choice as to the order in which a number of jobs can be processed. Especially in machine industry, they are quite important because the productibity is one of the objectives in machine shops and it may be significantly influenced by sequence. In practice there are many problems in a manufacturing organization in which the results of sequence are nontrivial and systematic consideration is worth while.[1] The chapters in this part of the dissertation are concerned mainly with exploring certain ways of finding sequences which are in some sense optimal.

Sequencing problems obviously get solved: Most of the problems are solved by intuition and experience without explicit recognition that a problem even existed.[2] It will be demonstrated in the following chapters, by analytical arguments and numerical examples, that there are significant differences between alternative sequences.

The difficulty in studying sequencing problems in applications is that the problems are not often independent and separable. They are interrelated with other types of decisions. If the problems are interrelated with the decisions as to what is to be done and how it is to be done, they are inextricably intractable. The solution is to neglect the possibility of changes in such decisions as what jobs have to be processed, the detailed character of the job, the procedure that will be used for performance and so forth, and consider only the simplified sequencing problems which attract theoretical attention. From a practical point of view such problems might be unrealistic in the sense that they do not represent any individual practical situations. But they do represent the way of thinking that should be followed in having a better insight into practical problems.

In what follows, the problems in which all the decisions relating to what and how have previously been made are considered provided that these decisions were entirely independent of the selection of sequence. To be more explicit, the following assumptions are made:

(1) All jobs to be performed are well defined and completely known, and are ready to start processing before the period under consideration begins.

(2) All the jobs must be performed.

(3) The facilities or the resources that are used in the execution of the jobs have been entirely specified.

(4) The method of performing each of the operations is entirely known and there is at least one of the set of facilities capable of performing the operations.

(5) A known and finite time is required to perform each

operation.

(6) Each operation, once started, must be performed to completion.

(7) Partially ordered technological restrictions, called precedence constraints, imposed on the operations are entirely known, including any restrictions on the order in which these must be performed.

The method of establishing sequences which satisfy various technological restrictions and how to select a suitable sequence from numerous feasible solutions are substantial to solve a sequencing problem. In machine industry there exist two characteristic types of sequences, classically identified as the linear sequence and the compound sequence. In the former, operations are performed one by one. While, in the latter a feasible subset of operations is assumed to be performed at a time.

Chapter 1 deals with the problem of establishing all of the feasible linear sequences which satisfy required precedence relationships. The essentials to analyze it are as follows:

(1) A suitable way of representing required precedence relationships.

(2) A systematical method of establishing all of the feasible linear sequences without overlapping.

(3) The total mumber of feasible linear sequences which satisfy required precedence relationships.

Chapter 2 is mainly concerned with the problems of defining a subset of the technologically feasible sequences that are preferred according to some criterion. There are three principal types of costs that can be affected by the decisions of pure sequence.[3] These are the costs of inventory, facility utilization, and lateness. Associated with these three basic cost factors, three problems are discussed in this chapter. These are the ones to sequence a set of jobs to minimize:

(1) Mean weighted flow-time with precedence restrictions.

(2) The sum of sequence-dependent setup-times with precedence restrictions.

(3) The total deferral cost associated with completion times with precedence restrictions.

The purpose of Chapter 3 is to find a systematic method to establish all of the feasible compound sequences which are composed of feasible subsets of operations.

Mainly the following are considered:

(1) A feasible subset of operations and its precedence relationships.

(2) The suitable way of constructing feasible subsets of operations without overlapping.

(3) The systematic way of establishing compound sequences.

Chapter 4 discusses the following two problems by the method of establishing compound sequences developed in Chapter 3:

(1) The problem of determining an optimum compound sequence which is composed of subsets of operations to minimize the sum of subset values associated with them with prececence restrictions.

(2) The line balancing problem.

The sequencing problems are essentially combinatorial and therefore the total number of feasible solutions is generally astronomical. Combinatorial problems might be characterized by the existence in most cases of a simply stated algorithm for enumerating all possible solutions, and by a factorial growth in the amount of computation required to carry out that enumeration as problem size increases. For particular problems it is possible to develop sophisticated enumerative methods that tend to produce satisfactory answers. In some cases it is even possible to find procedures to guarantee an optimum solution with a reasonable amount of computation. In what follows, effective algorithms to each of the above-mentioned problems will be developed after introduction of pertinent problem definition.

CHAPTER I ESTABLISHMENT OF LINEAR SEQUENCES

The purpose of this chapter is to find a systematic method to establish all of the feasible linear sequences which satisfy required precedence relationships. To answer the problem mainly the following are considered:

(1) A suitable way of representing required precedence relationships.

(2) A systematical method of establishing all of the feasible linear sequences without overlapping.

(3) The total number of feasible linear sequences which satisfy required precedence relationships.

After several definitions and notations are introduced, the first problem is tackled analytically in Section 1.2, and then based on the result of Section 1.2 the second problem is considered to develop sequential product and sequential multiplication in Section 1.3. There is no formula or prescription on hand to determine the total number of feasible linear sequences so far. Section 1.4 presents several methods for counting them which are answers to the third problem.

1. 1 PROBLEM DEFINITION

The basic unit of the production process is the operation. There is for each operation a partial ordering relationship with other operations by technological constraints or by some externally imposed policy. This partial ordering between operations is given by a binary relationship called precedence. If for some reason the processing of x_i must begin before the processing of x_j, then x_i is said to precede x_j and is written $x_i < x_j$.

The precedence relationship is transitive:

$$x_i < x_j \text{ and } x_j < x_k \quad \text{implies} \quad x_i < x_k.$$

A special case of this relationship exists when there are no intervening operations. It is said that operation x_i directly precedes operation x_j and is written $x_i \leqslant x_j$ if $x_i < x_j$ and there is no operation, x_k, such that $x_i < x_k < x_j$. Sometimes it is useful to use notation $x_i \leqslant x_j$ to mean that x_i directly or indirectly precedes x_j according to some reason.

It is often convenient to display required precedence relationships on a precedence graph.[1)] The nodes of the precedence graph represent the operations and the directed arrows represent "directly-preceds" relationships. The precedence relationship exists between two nodes if there is a path of directed arrows between them. It is assumed throughout that after ranking the nodes if the precedence diagram has more than one node whose rank is 1, an imaginary node should be added to the given precedence diagram to represent the beginning node. The precedence diagram whose nodes have been ranked already and which has only one beginning node is called the arranged precedence diagram.

The arranged precedence diagram can be represented by $G = [X, A]$, where,

$X = \{x_1, x_2, \ldots\ldots, x_n\}$: A set of the nodes, i.e., operations.

A: A subset of the arrows, i.e., the ordered pairs (x_i, x_j) taken from X, associated with required precedence relationships.

It is assumed throughout that X is a finite set, since the construction of computational procedures is mainly concerned. Notation $|X|$ represents the size of X, i.e., the number of elements forming X. A member, say, $a_{xp,xq}$ of A represents the arrow directing from node x_p to node x_q.

A linearly ordered arrangement of the nodes that can be performed in that order is called a feasible linear sequence. Now, n nodes can be arranged in n! distinct sequences. Because of required precedence relationships, only some of these n! will be feasible. Let $\pi(x_1.x_2\ldots\ldots x_{m-1}.x_m)$ or $(x_1, x_2)(\ldots)(x_{m-1}, x_m)$ denote a feasible linear sequence of distinct nodes of a precedence diagram. The notations may be shortened and are refered unambiguously to the sequence $x_1.\ x_2 \cdots x_{n-1}.x_n$ or π. Sometimes it is convenient to use notation $^m\pi_\nu(x_1, x_p)$ to represent the partial linear sequence of cardinal number m which passes m nodes from the beginning node x_1 to node x_p without breaking the precedence restrictions.

Since there are generally more than one sequence which passes m nodes from x_1 to x_p, ν should be added for distinction. ${}^m\pi$ (x_1, x_p) represents all of the linear sequences of cardinal number m which pass m nodes from x_1 to x_p :

$$^m\pi(x_1, x_p) = \bigvee_{\nu} {}^m\pi_\nu(x_1, x_p) ,$$

where $\bigvee$ is the notation of sequential union. In case the cardinal number covers all of the nodes, the sequence is called complete linear sequence. Let $S\{{}^m\pi_\nu(x_1, x_p)\}$ represent a set of the nodes which are contained in the sequence ${}^m\pi_\nu(x_1, x_p)$, and $\sigma\{{}^m\pi_\nu(x_1, x_p)\}$, a section of ${}^m\pi_\nu(x_1, x_p)$ by x_p :

$$\sigma\{{}^m\pi_\nu(x_1, x_p)\} = \{x_i | x_i \leqslant x_p, x_i \in S\{{}^m\pi_\nu(x_1, x_p)\}\}.$$

The purpose of this chapter is to find all of the feasible complete linear sequences for a given precedence diagram. They are nothing but Hamiltonian paths in a network diagram and may be obtained by using the multiplication-lathine of matrices. This procedure, unfortunately, leads to serious difficulties. First, the given precedence diagram must be re-written to the corresponding network diagram by adding nec-essary arrows since a transition is possible where there exists an arrow in the network diagram. By this operation the pre-cedence diagram may, however, become much more complicated and lose the merit of expressing conveniently required precedence relationships among the operations. Second, sequences which do not satisfy given precedence relationships may be produced with proper sequences, in obtaining feasible linear sequences from partial to complete step by step by the mulplication-lathine of matrices. This is a fatal defect for establishing feasible linear sequences systematically. The difficulties are due to the fact that the network diagram has been used for representing required precedence relationships without taking the essential differences into consideration between the precedence diagram and the network diagram. The precedence diagram has the following characteristics in comparison with the network diagram.

(1) It is a convenient display of precedence relationships, and does not indicate a path along which nodes should be passed.. In other words, as far as given precedence relationships are not disturbed transitions are possible whether arrows exists or not.

(2) All arrows are directed in one direction (generally from left to right).

A systematical method of establishing feasible linear sequences will be proposed in the following, based on the result of the analysis of these characteristics of the precedence diagram.

1.2 ANALYSIS OF PRECEDENCE RELATIONSHIPS AND INTRODUCTION OF A FUNDAMENTAL MATRIX

The basic method to establish feasible linear sequences will be to produce feasible linear sequences of high cardinal numbers from ones of lower by adding possible nodes to them without disturbing given precedence relationships. It will, first of all, be clarified how a sequence of high cardinal number can be produced from one of lower by one without breaking required precedence relationships. It will, then, be considered how to establish all of the feasible linear sequence systematically without overlapping.

It is necessary for the first purpose to analyse the precedence diagram which has the following characteristics as mentioned above : A path can be made by transition, if required precedence relationships are not disturbed whether arrows exists or not. This results in making the conditions clear under which a transition from one node to another is possible.

A matrix equivalent to the precedence diagram will, then, be introduced since the precedence diagram itself is not properly used for mathematical analysis. Up to the present various matrices equivalent to the precedence diagram are known, but these are not usable for establishing feasible linear sequences systematically, because these show only the possiblity of transition. For the above purpose, the conditions under which a transition is possible should be made clear, too. From this point of view, the conditions will be given under which a transition from one node to another directly is possible. After careful consideration they are summarized as follows:

I. If there exists an arrow α_{x_p, x_q} directing from node x_p to node x_q :

(1) In case there is only one arrow which directs to node x_q, i.e., α_{x_p, x_q}, the transition from x_p to x_q is possible.

(2) In case there are more than one arrows which direct to node x_q, say, $\alpha_{x_r, x_q}, \ldots, \alpha_{x_s, x_q}$, the transition from x_p to x_q is possible conditionally, i.e., under the conditions that all of the nodes $x_r, \ldots, x_s$ must be passed through before transiting from x_p to x_q.

II. If there exists no arrow directing from node x_p to node x_q :

(3) In case any nodes preceding x_p correspond to all of the nodes directing to x_q, the transition from x_p to x_q is possible.

(4) In case any nodes preceding x_p do not correspond to all of the nodes directing to x_q, the transition from x_p to x_q is possible conditionally, i.e., under the conditions that all of the nodes directing to x_q must be passed through before transiting from x_p to x_q.

The direct transition from x_p to x_q is not possible if none of (1), (2), (3), (4) above is applicable.

A matrix equivalent to the precedence diagram can be formed by making use of the above analysis. The matrix is called fundamental matrix.

Fundamental matrix : $L_o = (a_{p,q})$.

List the nodes of required precedence diagram vertically in ranking order, and then horizontally in the same order, and record $a_{p,q}$ in the column corresponding to the node q in the row corresponding to the node p, where,

$$a_{p,q} = \begin{cases} x_q : \text{In case the direct transition from } x_p \text{ to } x_q \text{ is possible.} \\ 0 : \text{Otherwise.} \end{cases}$$

Furthermore, the following c(p,q) must be indicated to give the information on conditional transition :

$$c(p,q) = \begin{cases} \phi : \text{In case of unconditional transition.} \\ \text{A set of the nodes that must be passed through in advance} \\ \quad : \text{In case of conditional transition.} \end{cases}$$

This c(p,q) will be written in a quarter-circle made at the northwestern corner of a box of each row and each column. Clearly, all information about the structure of a precedence diagram is embodied in its fundamental matrix. For example, the fundamental matrix for the precedence diagram shown in Fig.1.1 is given in Fig.1.2. In Fig.1.2 the box, say, at 7, column 8, has the information that the direct transition from node 7 to node 8 is possible after passing through nodes 2,5 and 6 in advance. Fig.1.3 shows into which categories [(1), (2), (3), (4)] the ways of direct transition enter in case direct transitions are possible.

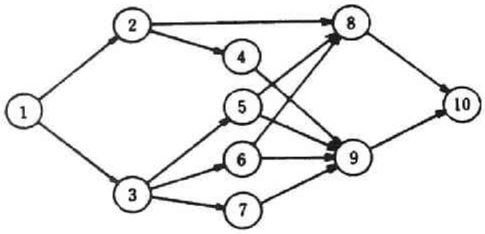


Fig. 1.1. Precedence diagram.

$L_0 =$

	1	2	3	4	5	6	7	8	9	10
1	0	2	3	0	0	0	0	0	0	0
2	0	0	3	4	^{3}5	^{3}6	^{3}7	^{5,6}8	0	0
3	0	2	0	^{2}4	5	6	7	0	0	0
4	0	0	3	0	^{3}5	^{3}6	^{3}7	^{5,6}8	^{5,6,7}9	0
5	0	2	0	^{2}4	0	6	7	^{2,6}8	^{4,6,7}9	0
6	0	2	0	^{2}4	5	0	7	^{2,5}8	^{4,5,7}9	0
7	0	2	0	^{2}4	5	6	0	^{2,5,6}8	^{4,5,6}9	0
8	0	0	0	4	0	0	7	0	^{4,7}9	^{9}10
9	0	0	0	0	0	0	0	8	0	^{8}10
10	0	0	0	0	0	0	0	0	0	0

Fig. 1.2. The fundamental matrix.

	1	2	3	4	5	6	7	8	9	10
1		1	1							
2			3	1	4	4	4	2		
3		3		4	1	1	1			
4			3		4	4	4	4	2	
5		3		4		3	3	2	2	
6		3		4	3		3	2	2	
7		3		4	3	3		4	2	
8				3			3		4	4
9								3		4
10										

Fig. 1.3. Classification of direct transitions.

By arranging precedence relationships with the concept of direct transition, it becomes much easier to handle them analytically.

1. 3 SEQUENTIAL PRODUCT AND SEQUENTIAL MULTIPLICATION OF MATRICES

1. 3. 1 Sequential Product

The next problem to be discussed will be to clarify the process how a feasible linear sequence of high cardinal number is made from a feasible linear sequence of lower one. For this purpose remember the difficulties that have been encountered when the network diagram is adopted in order to establish feasible linear sequences systematically. That is, consider the following to prevent from producing infeasible linear sequences :

(1) Not to disturb required precedence relationships.

(2) To prohibit to get a cycle.

The former (1) will be settled by making use of the fundamental matrix introduced in Section 1.2 since it has all the information on direct transition. The latter (2) is easy to be settled. Taking these into consideration and arranging the process of establishing a feasible linear sequence from lower to higher with the fundamental matrix, results in the definition of sequential product whose flow chart is shown in Fig. 1.4.

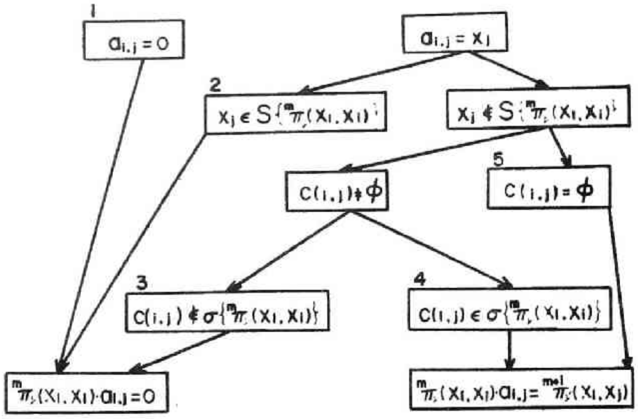


Fig. 1.4. Sequential product.

In the Figure, if the direct transition from x_i of ${}^m\pi_\nu(x_1, x_i)$ to x_j is not possible, the product of ${}^m\pi_\nu(x_1, x_i)$ and $a_{i,j}$ leads to zero as shown in 1. If the direct transition from x_i to x_j is possible, i.e., $a_{i,j} = x_j$, and if x_j is contained in $S\{{}^m\pi_\nu(x_1, x_i)\}$, then the product also leads to zero to prohibit a cycle as shown in 2. If x_j is not contained in $S\{{}^m\pi_\nu(x_1, x_i)\}$ and if $c(i,j)$ is not contained in $\sigma\{{}^m\pi_\nu(x_1, x_i)\}$ then the product leads to zero to exclude a sequence which violates required precedence relationships. A new higher feasible linear sequence is obtained if 4 or 5 is satisfied. The operation is called sequential product. Sequential product makes it possible to excludes all the difficulties that have been encountered.

1.3.2 Sequential Multiplication of Matrices

The basic operations for establishing a feasible linear sequence have been settled so far. The next problem to be discussed is to introduce a proper way of establishing all of the feasible linear sequences systematically. For this purpose the operation of multiplication of matrices may be useful.

Definition of sequential multiplication of matrices:

$$[M]^{m+1} = [M]^{m} \cdot L_o \ . \tag{1.1}$$

The product of a $(1,n)$ matrix $[M]^m$ by an (n,n) fundamental matrix L_o is a $(1,n)$ matrix $[M]^{m+1}$ whose element in row 1, column j is the sequential product of $[M]^m$ by the j th column of L_o . To illustrate, let

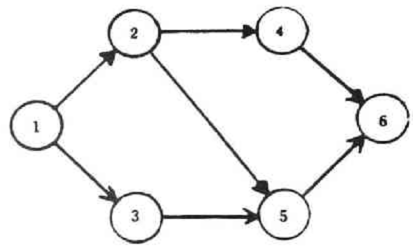
$$[M]^m = \begin{array}{cccccccc} & x_1 & x_2 & \cdots & x_i & \cdots & x_n \\ x_1 & [\,0 & {}^m\pi(x_1,x_2) & \cdots & {}^m\pi(x_1,x_i) & \cdots & {}^m\pi(x_1,x_n)\,] \end{array} . \tag{1.2}$$

Then, by definition,

$${}^{m+1}\pi(x_1,x_j) = \bigvee_{\nu'} {}^{m+1}\pi_{\nu'}(x_1,x_j) = \bigvee_{i=1}^{n} (\bigvee_{\nu} {}^{m}\pi_{\nu}(x_1,x_i) \cdot a_{i,j}). \tag{1.3}$$

The row of the beginning node is adopted as $[M]^1$, and the matrix, as $[M]^2$, whose element in row 1, column j is the sequential product of the elements of the beginning node by $[M]^1$.

It is possible to obtain feasible linear sequences of high cardinal number by the equation (1,1) and finally complete linear sequences by $[M]^n$ if $|X| = n$.



(a) Precedence diagram.

	1	2	3	4	5	6
1	0	2	3	0	0	0
2	0	0	3	4	$^{3}5$	0
3	0	2	0	$^{2}4$	$^{2}5$	0
4	0	0	3	0	$^{3}5$	$^{5}6$
5	0	0	0	4	0	$^{4}6$
6	0	0	0	4	0	0

(b) Fundamental matrix.

$$[M]^1 = [\; 0 \quad 2 \quad 3 \quad 0 \quad 0 \quad 0 \;]$$

$$[M]^2 = [\; 0 \quad 1\cdot2 \quad 1\cdot3 \quad 0 \quad 0 \quad 0 \;]$$

$$[M]^3 = [\; 0 \quad 1\cdot3\cdot2 \quad 1\cdot2\cdot3 \quad 1\cdot2\cdot4 \quad 0 \quad 0 \;]$$

$$[M]^4 = \left[\; 0 \quad 0 \quad 1\cdot2\cdot4\cdot3 \quad \begin{matrix} 1\cdot3\cdot2\cdot4 \\ 1\cdot2\cdot3\cdot4 \end{matrix} \quad \begin{matrix} 1\cdot3\cdot2\cdot5 \\ 1\cdot2\cdot3\cdot5 \end{matrix} \quad 0 \;\right]$$

$$[M]^5 = \left[\; 0 \quad 0 \quad 0 \quad \begin{matrix} 1\cdot3\cdot2\cdot5\cdot4 \\ 1\cdot2\cdot3\cdot5\cdot4 \end{matrix} \quad \begin{matrix} 1\cdot2\cdot4\cdot3\cdot5 \\ 1\cdot2\cdot3\cdot4\cdot5 \\ 1\cdot3\cdot2\cdot4\cdot5 \end{matrix} \quad 0 \;\right]$$

$$[M]^6 = \left[\; 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \begin{matrix} 1\cdot3\cdot2\cdot5\cdot4\cdot6 \\ 1\cdot2\cdot3\cdot5\cdot4\cdot6 \\ 1\cdot2\cdot4\cdot3\cdot5\cdot6 \\ 1\cdot2\cdot3\cdot4\cdot5\cdot6 \\ 1\cdot3\cdot2\cdot4\cdot5\cdot6 \end{matrix} \;\right]$$

(c) The steps by which five complete linear sequences are produced.

Fig. 1.5. Illustration of sequential multiplication.

For the precedence diagram as shown in Fig. 1.5(a) whose fundamental matrix is given in Fig. 1.5(b), for example, five complete linear sequences will be obtained by the steps in Fig. 1.5(c). In the next section it will be clarified that there are no more feasible linear sequences other than these five sequences for the given precedence diagram, but it may be obvious from the definition of sequential product.

The characteristics of sequential multiplication are that feasible linear sequences of high cardinal number can easily be established successively without producing infeasible linear sequences since the product of an element of a matrix by an element of a fundamental matrix follows sequential product and that feasible linear sequences are made out systematically without overlapping by the operation of sequential multiplication of matrices.

1.4 THE TOTAL NUMBER OF FEASIBLE COMPLETE LINEAR SEQUENCES

It will be discussed as to how many feasible linear sequences there are in relation to the problem of establishing feasible linear sequences satisfying required precedence relationships. This problem is not necessarily a fundamental problem for the sequencing problem, but this total number may give a good measure while performing calculation.

Klein[2), Ignall[3) and others have considered the same kind of problem, but did not propose any effective method.

The basic method will be to apply combinatorial theory to the problem by dividing required precedence relationships into groups and rearranging them in such a way that the number of feasible linear sequences in each group can be calculated more easily. From this point of view, one basic method will be introduced. It is called the box method.

1.4.1 The Box Method

This method seeks the total number of feasible complete linear sequences by making as many boxes as the number of the nodes and considers the ways of putting them into the boxes without disturbing required precedence relationships. Elementary lemmas used for the method are as follows.

Lemma 1.1. In a set of nodes, assume that the original n nodes have been grouped into k strings with h_s, i_t, ..., j_u as the number of nodes in the various strings :

$$\left.\begin{array}{ll} P: & x_{h_1} \leqslant x_{h_2} \cdots \leqslant x_{h_s} \\ Q: & x_{i_1} \leqslant x_{i_2} \cdots \leqslant x_{i_t} \\ \vdots & \quad \cdots\cdots \\ R: & x_{j_1} \leqslant x_{j_2} \cdots \leqslant x_{j_u} \end{array}\right\} \quad \text{where, } s+t+\cdots+u \leq n.$$

Assume that the membership of each string is fixed, and that P, Q, ... , R are mutually exclusive, that is, no two have a node in common. Then the total number of feasible linear sequences N is

$$N = \frac{n!}{s!\ t!\ \ldots\ u!} \quad . \tag{1.4}$$

(Proof) Self-evident.

Lemma 1.2. Divide all of the nodes in X into the fixed order nodes that must be performed in specific locations and the variable order nodes that may be done in any location :

Fixed order nodes: $x_{k1}, \ldots, x_{kv}$ where,

Variable order nodes: $x_{e1}, \ldots, x_{ew}$ $v + w = n$.

Then, the total number of feasible linear sequences N is

$N = {}_nC_v \cdot (n-v)! \cdot$ (the possible number of feasible sub-sequences among the fixed order nodes) . (1.5)

(Proof) There are ${}_nC_v$ kinds of ways of putting the fixed order nodes into the boxes, and (n-v)! kinds of ways for the variable order nodes since the order of the set is quite unrestricted. Also some possible combinations exist among the fixed order nodes. Taking these into consideration, results in the above equation (1.5).

The method is applicable to the cases where required precedence relationships can be divided into some mutually exclusive groups by proper ingenuity. If the given precedence relationships are too complicated to be divided into groups, some supplementary methods should be used. Some of them will be developed in the following.

1.4.2 The Fixing Method

The method partitions the given complicated precedence relationships into several groups, which have simpler precedence relationships, by fixing a certain node to a specific position of the boxes. As a fixing node, the node that is going to make given precedence relationships simpler by its fixing or the one that has a small number of ways of putting it into the boxes is to be selected.

For example, the total number of feasible linear sequences for the precedence diagram shown in Fig. 1.1 is calculated by the method: Following the box method ten boxes are made. Then node 1 enters the first box, and node 10 the last box. If attention is fixed to node 9, this enters the eighth or the ninth box. Divide the precedence relationships into two cases. There are 90 feasible complete linear sequences in the first case in which node 9 enters the eighth box and 156 in the second case in which node 9 enters the ninth box. Since there are no linear sequences in common between the first 90 linear sequences and the second 156 linear sequences, there are totally 246 feasible complete linear sequences.

However complicated required precedence relationships are, the method is quite effective for calculating the total number of feasible linear sequences.

1.4.3 The Inverse Arrow Method

This method abbreviates for a while a certain arrow, say, $\alpha_{x_p},\ x_q$ which is rendering the box method unsuitable, and seeks the number of complete linear sequences $N(\alpha)$ in the case where there is no arrow between x_p and x_q. Among $N(\alpha)$ complete linear sequences, let the number of linear sequences satisfying the condition $x_p \leqslant x_q$ be denoted by $N(\alpha x_p, x_q)$, and the number of linear sequences satisfying the condition $x_q \leqslant x_p$ by $N(\alpha_{x_q},\ x_p)$. Then,

$$N(\alpha)=N(\alpha_{x_p},\ x_q) + N(\alpha_{x_q},\ x_p).$$

Therefore,

$$N(\alpha_{x_p},\ x_q)=N(\alpha)-N(\alpha_{x_q},\ x_p). \qquad (1.6)$$

In some cases of precedence diagrams, $N(\alpha_{x_q},\ x_p)$ might be more easily calculated than $N(\alpha_{x_p},\ x_q)$. In such cases the method is quite effective. Even if it is necessary to abbreviate two or more arrows, the method is still applicable. Abbreviating two arrows, say, $\alpha_{x_p},\ x_q$ and $\alpha_{x_r},\ x_s$, results in the following:

$$N(\alpha_{x_p},x_q\cdot\alpha_{x_r},x_s)=N(\alpha)-N(\alpha_{x_q},x_p)-N(\alpha_{x_s},x_r)+N(\alpha_{x_q},x_p,\alpha_{x_s},x_r), (1.7)$$

where, as far as the notations in the above equation (1.7) are concerned, they are assumed to be easily understood.

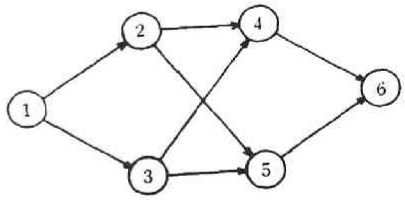


Fig. 1.6. Precedence diagram.

For example, abbreviating arrows $\alpha_{3,4}$ and $\alpha_{2,5}$ for the precedence diagram shown in Fig. 1.6, results in

$N(\alpha)=6$, $N(\alpha_{4,3})=N(\alpha_{5,2})=1$, $N(\alpha_{4,3}\cdot\alpha_{5,2})=0$.

Therefore,

$N(\alpha_{3,4}\cdot\alpha_{2,5}) = 6-1-1+0 = 4$.

This is the total number of feasible complete linear sequences for the precedence diagram Fig. 1.6.

1.4.4 The Grouping Method

From an analytical point of view it is often convenient and useful to divide the given nodes into several groups. The method developed here is applicable to such a case. It tries to make given precedence relationships simpler by dividing the nodes into several subgroups without breaking the precedence relationships and seeks the number of complete linear sequences satisfying the simpler precedence relationships, and then counts the complete linear sequences lost by dividing into groups. It is assumed that a transition from one subgroup to another is possible only when it is not breaking the precedence relationship between two subgroups and only after all of the nodes of the former subgroup are passed through.

Under this assumption, the following situations occur: If there exists an arrow from a node, say, x_p, to a node, say, x_q in a subgroup, a direct transition from node x_p to any node in other subgroups is not possible. Moreover a direct transition from any node in other subgroups to node x_q is not possible . These facts are used to count the number of complete linear sequences lost by grouping. Attention should be paid to the following: According to improper grouping, no complete linear sequences may exist. In simple cases it can be judged from personal observation whether the grouping is proper or not, but in complicated cases, the following may be useful: Suppose that a set of the nodes X are divided into m subgroups $W_1, W_2, \ldots, W_m$. Make an (m x m) matrix whose element in row i, column j is 1 in case there exists at least one arrow directing from a node of W_i to a node of W_j, and 0 otherwise. Then, rank subgroups $W_1, W_2, \ldots, W_m$ by the matrix. If there are no cycles, the grouping is proper and complete linear sequences do exist.

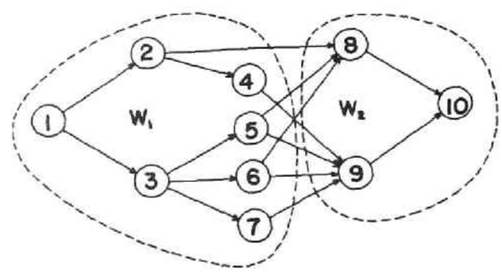


Fig. 1.7. Precedence diagram.

$M_1 =$

	1	2	3	4	5	6	7
1	0	2	3	0	0	0	0
2	0	0	3	4	$^{(3)}5$	$^{(3)}6$	$^{(3)}7$
3	0	2	0	$^{(2)}4$	5	6	7
4	0	0	3	0	$^{(3)}5$	$^{(3)}6$	$^{(3)}7$
5	0	2	0	$^{(2)}4$	0	6	7
6	0	2	0	$^{(2)}4$	5	0	7
7	0	2	0	$^{(2)}4$	5	6	0

$M_{1,2} =$

	8	9	10
1	0	0	0
2	$^{(5,6)}\cancel{8}$	0	0
3	0	0	0
4	$^{(5,6)}8$	$^{(5,6,7)}9$	0
5	$^{(2,6)}8$	$^{(4,6)}9$	0
6	$^{(2,5)}8$	$^{(4,5,7)}9$	0
7	$^{(2,5,6)}8$	$^{(4,5,6)}9$	0

$M_{2,1} =$

	1	2	3	4	5	6	7
8	0	0	0	$\cancel{4}$	0	0	$\cancel{7}$
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0

$M_2 =$

	8	9	10
8	0	$^{(4,7)}9$	$^{(9)}10$
9	8	0	$^{(8)}10$
10	0	0	0

Fig. 1.8. Fundamental matrix.

The grouping method is applied, for example, to the precedence diagram shown in Fig. 1.1. Let the precedence diagram be divided into two subgroups W_1, W_2 as shown in Fig. 1.7. In this case there is a precedence relationship $W_1 \ll W_2$ between the two subgroups. The number of complete linear sequences satisfying the resulting precedence relationships including $W_1 \ll W_2$ is as follows :

$$N_1 = 90 \times 2 = 180 .$$

The fundamental matrix of this case is shown in Fig. 1.8, where,

M_1 : The fundamental matrix of subgroup W_1 .

M_2 : The fundamental matrix of subgroup W_2 .

$M_{1,2}$: The transition matrix from W_1 to W_2 .

$M_{2,1}$: The transition matrix from W_1 to W_2 .

The arrows $\alpha_{8,4}$ and $\alpha_{8,7}$ in the transition matrix $M_{2,1}$ and $\alpha_{2,8}$ in $M_{1,2}$ get lost by the grouping. Let

N_2 : The number of complete linear sequences including the direct transition from node 2 to node 8.

N_3 : " " " from node 8 to node 4.

N_4 : " " " from node 8 to node 7.

$N_{2,3}$: " " the direct transitions from node 2 to node 8 and from node 8 to node 4.

$N_{2,4}$: " " " " from node 8 to node 4 and from node 2 to node 8.

$N_{3,4}$: " " " " from node 8 to node 7 and from node 8 to node 4.

$N_{2,3,4}$: " " " " from node 2 to node 8, from node 8 to node 4 and from node8 to node 7.

Then, the total number of complete linear sequences for the precedence diagram Fig. 2.1 is

$$N = N_1+N_2+N_3+N_4-N_{2,3}-N_{2,4}-N_{3,4}+N_{2,3,4}$$

$$= 180+10+38+28-8-2-0+0 = 246 \ .$$

This number corresponds with the one sought previously. The lost sequences by the grouping is 27% of all sequences.

1.5 CONCLUSIONS

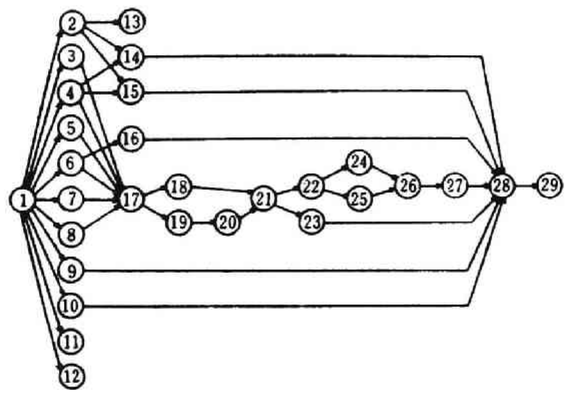


Fig. 1.9. Precedence diagram.

An application of the above methods to the more practical precedence diagram shown in Fig. 1.9 results in 1, 762, 551, 451, 584 ($\fallingdotseq 1.8 \times 10^{12}$) complete linear sequences.

The precedence relationships reduce the $29!$($\fallingdotseq 8.8\times10^{30}$) possible permutations to about 1.8×10^{12} feasible complete linear sequences which are about 2.0×10^{-19} of all the permutations.

This indicates that it is quite important to establish feasible sequences without producing any improper sequence and systematically. As sequential multiplications never count the sequences which do not satisfy required precedence relationships, it may be well understood how effective and useful sequential multiplications are for establishing feasible linear sequences.

CHAPTER 2 DECISION OF OPTIMUM LINEAR SEQUENCES

2.1 Three Problems

This chapter is mainly concerned with the case of the sequencing problem in which each job consists of a single operation. Since the set of jobs can be partitioned depending on the machine required to perform the operation, each machine in the shop is independent of the others and can be scheduled separately. Therefore attention can be limited to a single machine, and to the set of jobs to be processed on that machine.

The sequencing problem of single machine systems is not only the base for solving more complex sequencing problems, but also is a quite interesting problem in itself. From a practical point of view this model will find direct applications frequently. Mechanical manufacturing industries, for instance, with increasing automation are headed in this direction. As the use of automatic-control equipment and complex transfer machines increases, production facilities tend to a system which is, or is predominated by, a single unit of equipment. There are also situations in which, although each job actually consists of several operations to be performed on different machines, there is one particular machine whose value and function so dominates the process that it is sequenced as if the other machines did not exist.

It will be assumed throughout this chapter that the number of jobs n is finite, that n is known in advance of sequencing, and that all the n jobs must be processed. Furthermore, it will be assumed that the machine is to have no other obligations and that it will be continuously available, without breakdown, until all the jobs are completed. The processing-times are arbitrary, determined by some process independent of the scheduling procedure, and are assumed known at the time of scheduling.

The costs directly associated with the sequencing problem of single machine systems are restricted. It is assumed throughout this chapter that the method and the efficiency with which they will be employed are unaffected by scheduling decisions. The assumption is not unnatural since the costs that may be attributed to decisions of pure sequence are usually what would be classified as facility costs rather than product costs.

There are three principal types of costs that can be affected by the decisions of pure sequence. These are the costs of inventory, facility utilization, and lateness.

In general it can be said that there are strong economic reasons in reducing average inventories. Under the above assumptions, the costs of inventory provide the incentive for minimizing average flow-time, since they are directly related to average inventory. In Section 2.2 the problem to sequence a set of jobs to minimize mean weighted flow-time with precedence restrictions will be tackled. It will be assumed in the section that all the n jobs are available for processing simultaneously so that the schedule could begin with any one of them, and that there is either no setup-time required for the jobs, or that the setup-time does not depend on the nature of the preceding job on the machine. In the latter case the setup-time for a particular job depends only on the characteristics of that job and may, for present purposes, be included in the processing-time for the job.

Facility utilization is a very important economic consequence of sequencing decisions. The ability to compact the busy intervals and produce a short schedule-time simply implies a procedure that will permit a given work load to be accomplished with a smaller aggregate demand on facilities. The costs of facility utilization provide the incentive for minimizing the total amount of time required to process all the jobs. Where setup is assumed to be independent of sequence as in Section 2.2, setup-time can be included in the processing time, and therefore the total amount of time required is a constant, independent of sequence. While in the cases in which the setup-time is sequence dependent, the circumstances are considerably altered as compared with the case of Section 2.2 and the total amount of time required depends on how the jobs are ordered. The sum of the processing-times is still a constant, and may be ignored; the maximum flow-time is minimized by minimizing the sum of the n setup-times. In fact, there are many situations in which it is simply not acceptable to assume that the time required to set up the machine for the next job is independent of the job that was the immediate predecessor on the machine. In Section 3.3 the problem of sequencing a set of jobs to minimize the sum of sequence-dependent setup-times with precedence restrictions will be tackled. In the section the assumption that the jobs arrive simultaneously will not be required.

In some situations, especially construction projects, the cost of lateness is obvious and explicit. In Section 2.4 the problem to sequence a set of jobs to minimize the total deferral cost associated with completion times with precedence restrictions will be considered. The assumptions introduced in Section 2.4 will be required in the section.

2.2 SEQUENCING A SET OF JOBS TO MINIMIZE MEAN WEIGHTED FLOW-TIME WITH PRECEDENCE RESTRICTIONS

2.2.1 Problem Statement

The jobs to be processed are identified by the integers 1, 2, ..., n. The relevant attributes of job i that are given as part of the problem description are denoted by the following variables :

p_i is the processing-time, the amount of time that will be required to perform the job i .

u_i (>0) is given for each job to describe relative importance and used as coefficient in performance measures.

The problem is to minimize mean weighted flow-time without violating required precedence restrictions :

$$F = \frac{1}{n} \operatorname{Min} \sum_{i=1}^{n} u_i t_i \quad , \qquad (2.1)$$

where t_i is the flow-time of job i : the total time that the job spends in the shop.

Generally seeking an optimum solution for a combinatorial problem requires a troublesome caluculation of the order of factorial. It is desired by the discovery of useful theorems to reduce the calculation to exponential order, hopefully, to polynomial order. The above problem has been a matter of academic concern in the last two decades. The special case in which precedence restrictions are not imposed was solved by polynomial order by Mcnaughton (1959)[1], and Smith (1956)[2]. Gapp, Mankekar, Mitten (1965)[3], and Bowden (1969)[4] have tried to solve the case in which precedence restrictions are imposed, but no effective method has been found so far in the case where arbitrary precedence relationships are imposed. The purpose of this section is to develop an effective algorithm for solving the general case which permits the existence of arbitrary precedence restrictions.

2.2.2 Analysis

The most proper approach to be taken in the first place, in order to solve a combinatorial problem, if a special effective method cannot be found, is to try to reduce by some methods the existential range of an optimum solution. It will be considered in the following what methods are effective to reduce the existential range of an optimum solution

In the case in which precedence relationships are not imposed, McNaughton[1), and Smith[2) have shown that mean flow-time is minimized by sequencing jobs according to the ratio p_i/u_i, with the job having the smallest ratio being performed first.

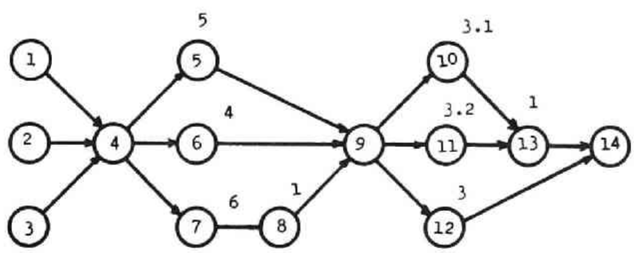


Fig. 2.1. Precedence diagram.

Let this theorem apply to the case in which precedence relationships are imposed, for example, to the precedence diagram shown in Fig. 2.1. The numbers inside the circles identify jobs. Job 4 is a break point at which the problem can be partitioned. Jobs 1, 2, and 3 are effectively independent, and therefore the above theorem applies. Job 9 is also a break point, and so jobs 5, 6, and 8 which precede job 9 are considered. The processing times of jobs 5, 6, 7, and 8 are shown outside the circles. The relative importance of each job is assumed to be 1. The first job to be handled is 5, 6, or 7. According to the above theorem, job 6 is performed first since job 6 has the smallest ratio p_i / u_i among jobs 5, 6, and 7. Then, jobs 5, 7, and 8 are processed in this order. Putting jobs 5, 6, 7, and 8 in executable order, results in twelve feasible sequences, among these sequences, $7 \cdot 8 \cdot 6 \cdot 5$ gives the minimum mean weighted flow-time instead of sequence $6 \cdot 5 \cdot 7 \cdot 8$ obtained above by the application of the theorem. This indicates that the theorem is not applicable to such a case as in Fig. 2.1. On second thoughts it is noticed that job 8 has the smallest processing-time among jobs 5, 6, 7, and 8, and therefore it should be processed as early as possible. The optimum sequence is surely the feasible sequence which processes job 8 as early as possible. Let jobs 7 and 8 be combined into one job denoted 7 8 and let its processing-time be the sum of the processing-times of job 7 and job 8. Also let the sum of the relative importance of job 7 and job 8 be that of job 7 8. Then the ratios of jobs 7 8 , 5, and 6 become larger in this order. The theorem by Smith is applicable to this case, and an optimum sequence 7 8 · 5 · 6 which corresponds to the optimum sequence obtained before will be obtained.

It is surmised from the above analysis that $\Sigma(p_i)/\Sigma(u_i)$ plays more important role than the individual job ratio p_i/u_i, when precedence restrictions are imposed. The job or the set of jobs to be performed in the first place among jobs are 5, 6, 7, or 7 · 8 . To consider $\Sigma(p_i)/\Sigma(u_i)$ generally, let

$\delta_\nu(i)$: A set of job i and jobs which follow i . Since there are usually more than one set which satisfies this condition, ν should be added for distinction. For example,

$\delta_1(7) = 7,\quad \delta_2(7) = 7 \cdot 8.$

$r_{i\nu} := \sum_{j\in\delta_\nu(i)} (p_i) / \sum_{j\in\delta_\nu(i)} (u_j)$. For example,

$$r_{71} = \sum_{j\in\delta_1(7)} (p_j) / \sum_{j\in\delta_1(7)} (u_j) = 6/1 = 6 \quad ,$$

$$r_{72} = \sum_{j\in\delta_2(7)} (p_j) / \sum_{j\in\delta_2(7)} (u_j) = (6+1)/(1+1) = 3.5 .$$

$r_i^* := \underset{\nu}{\mathrm{Min}}\{r_{i_\nu}\}$. For example, $r_7^* = \mathrm{Min}\{r_{7_1}, r_{7_2}\} = 3.5$.

Following these notations, it can be said that in case the set of jobs can be divided without disturbing required precedence relationships into mutually exclusive independent subset of jobs, mean weighted flow-time is minimized by sequencing the jobs according to the ratio r_i * with the set of jobs having the smallest ratio being performed first. For the purpose of giving a proof to this, the characteristics of r_i * will be investigated forst.

Theorem 2. 1. In the set of jobs that constitute a chain which occurs when precedence constraints give each job at most one predecessor and at most one successor, compute r_i * (i is the first job in the chain), and then r_j *(j is the first job in the resulting chain after deleting the jobs constituting r_i * from the original chain), etc., then,

$$r_i * \leq r_j * \leq \cdots .$$

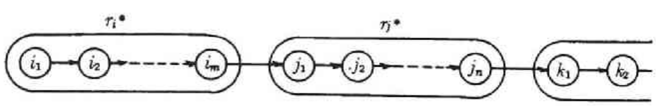


Fig. 2.2. A chain.

(Proof) The chain can conveniently be shown as in Fig. 2.2. Suppose that jobs i_1, i_2, ... , and i_m constitute r_i *, and jobs j_1, j_2, ..., and j_n, r_j * as shown in Fig. 2.2. From the definition of r_i *,

$$r_i^* = \frac{\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{m} u_{iq}} = \mathrm{Min}\left[\frac{p_{i_1}}{u_{i_1}}, \frac{p_{i_1}+p_{i_2}}{u_{i_1}+u_{i_2}}, \cdots, \frac{\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{m} u_{iq}}, \frac{\sum_{q=1}^{n} p_{iq}+p_{j_1}}{\sum_{q=1}^{n} u_{iq}+u_{j_1}}, \right.$$

$$\left. \frac{\sum_{q=1}^{m} p_{iq}+p_{j_1}+p_{j_2}}{\sum_{q=1}^{m} u_{iq}+u_{j_1}+u_{j_2}}, \cdots, \frac{\sum_{q=1}^{m} p_{iq}+\sum_{q=1}^{n} p_{jq}}{\sum_{q=1}^{m} u_{iq}+\sum_{q=1}^{n} u_{jq}}, \cdots \right].$$

Therefore,

$$r_i^* = \frac{\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{m} u_{iq}} \leqq \frac{\sum_{q=1}^{m} p_{iq}+\sum_{q=1}^{n} p_{jq}}{\sum_{q=1}^{m} u_{iq}+\sum_{q=1}^{n} u_{jq}} \quad . \tag{2.2}$$

To put the above inequality into an equation, let

$$k \frac{\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{m} u_{iq}} = \frac{\sum_{q=1}^{m} p_{iq}+\sum_{q=1}^{n} p_{jq}}{\sum_{q=1}^{m} u_{iq}+\sum_{q=1}^{n} u_{jq}} \quad . \tag{2.3}$$

Then , $k \geqq 1$.

Since $a/b = c/d$ (where $b \neq d$) implies $c/d = (c-a)/(d-b)$ without loss of generality,

$$\frac{\sum_{q=1}^{m} p_{iq}+\sum_{q=1}^{n} p_{jq}}{\sum_{q=1}^{m} u_{iq}+\sum_{q=1}^{n} u_{jq}}=\frac{\sum_{q=1}^{m} p_{iq}+\sum_{q=1}^{n} p_{jq}-k\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{m} u_{iq}+\sum_{q=1}^{n} u_{jq}-k\sum_{q=1}^{m} u_{iq}}=\frac{\sum_{q=1}^{n} p_{jq}}{\sum_{q=1}^{n} u_{jq}}-(k-1)\frac{\sum_{q=1}^{m} p_{jq}}{\sum_{q=1}^{n} u_{jq}} \qquad (2.4)$$

From (2.2), (2.3), and (2.4),

$$r_i^* \leq \frac{\sum_{q=1}^{m} p_{iq}+\sum_{q=1}^{n} p_{jq}}{\sum_{q=1}^{m} u_{iq}+\sum_{q=1}^{n} u_{jq}}=\frac{\sum_{q=1}^{n} p_{jq}}{\sum_{q=1}^{n} u_{jq}}-(k-1)\frac{\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{n} u_{jq}} \leq \frac{\sum_{q=1}^{n} p_{jq}}{\sum_{q=1}^{n} u_{jq}} \leq r_j^* \; .$$

Although Theorem 2.1 describes a characteristic of r_i * of the jobs which constitute a chain, it applies to the case in which the given jobs have precedence relationships of tree-type, as shown in Fig 2.3, which occur where precedence constraints give each job at most one predecessor. The proof of this case is almost the same as above and therefore is omitted.

Theorem 2.2. When the jobs constituting r_i * is divided into the preceding part and the following part,

r_i * (the following part) $\leq r_i^* \leq r_i^*$ (the preceding part),

where r_i * (the following part) is the total sum of the processing-times of the following part divided by the sum of the relative importance of each job of the following part, and r_i * (the preceding part) is the same value of the preceding part.

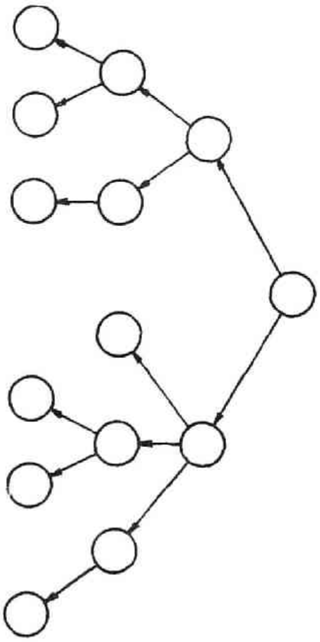


Fig. 2.3. Precedence diagram of tree type.

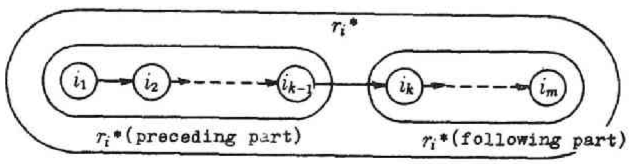


Fig. 2.4. The jobs constituting r_i* .

(Proof) Suppose r_i * is constituted by jobs i_1, i_2, ..., i_{k-1}, i_k, ... , i_n, as shown in Fig. 2.4. Let the jobs be divided into i_i, i_2, ... , i_{k-1} (the preceding part), and i_k, ... , i_m (the following part). Then,

$$\frac{\sum_{q=k}^{m} p_{iq}}{\sum_{q=k}^{m} u_{iq}} \leq \frac{\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{m} u_{iq}} \leq \frac{\sum_{q=1}^{k-1} p_{iq}}{\sum_{q=1}^{k-1} u_{iq}}$$

is to be shown. The latter part of the above inequalities is self-evident from the definition of r_i * . To put the former part into an equation, let

$$\frac{\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{m} u_{iq}} = l \frac{\sum_{q=1}^{k-1} p_{iq}}{\sum_{q=1}^{k-1} u_{iq}} ,$$

then $\qquad 0 < l \leq 1 \quad .$

Therefore

$$\frac{\sum_{q=1}^{m} p_{iq}}{\sum_{q=1}^{m} u_{iq}} = \frac{\sum_{q=1}^{m} p_{iq} - l \sum_{q=1}^{k-1} p_{iq}}{\sum_{q=1}^{m} u_{iq} - \sum_{q=1}^{k-1} u_{iq}} = \frac{\sum_{q=k}^{m} p_{iq} + (1-l) \sum_{q=1}^{k-1} p_{iq}}{\sum_{q=k}^{m} u_{iq}}$$

$$= \frac{\sum_{q=k}^{m} p_{iq}}{\sum_{q=k}^{m} u_{iq}} + (1-l) \frac{\sum_{q=1}^{k-1} p_{iq}}{\sum_{q=k}^{m} u_{iq}} \geqq \frac{\sum_{q=k}^{m} p_{iq}}{\sum_{q=k}^{m} u_{iq}} .$$

After the above prerarations, it will be shown by Theorem 2.3 that r_i * plays an important role in case precedence relationships are imposed on a set of jobs.

<u>Theorem 2.3.</u> In the case in which n jobs are divided into

p chains within which job order is specified, but which may be preempted between jobs, compute r_i * in each chain, and select the job or the set of jobs which gives the minimum ratio r_i * (denoted r_1 *), and then delete it from the original chains. In the resulting chains, continue the same procedure and obtain r_2 *, etc. Then the mean weighted flow-time can be minimized by sequencing the job or the set of jobs constituting r_1 * first and then r_2 *, etc.

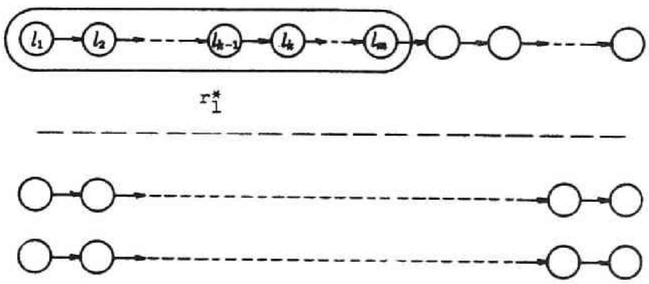


Fig. 2.5. Chains.

(Proof) Note that $r_1{*} \leq r_2{*} \leq \cdots$ from Theorem 2.1. Suppose that Fig. 2.5 shows given p chains and that jobs $l_1, l_2, \cdots, l_{k-1}, l_k, \cdots, l_m$ constitute r_1 *. Suppose that by some procedure a feasible sequence S_α has been obtained already, and let S_β be the sequence obtained from changing the order of the jobs in S_α in such a way that the set of the jobs constituting r_1 * comes in the first place and the order of the other jobs is the same as S_α. Since the total number of jobs to be performed is fixed, minimizing mean weighted flow-time is equivalent to minimizing total weighted flow-time, the difference of the total weighted flow-times between S_α and S_β is considered. In what follows, The total weighted flow-time of $S_\alpha \geqslant$ the total weighted flow-time of S_β is to be shown. For example, Fig. 2.6 shows that sequence S_α has the set of jobs $i_1, i_2, \cdots, i_n$ first; the preceding part of the jobs constituting r_1 *, l_1, l_2, $\cdots$, and l_{k-1}, second; $j_1, j_2, \cdots, j_n$ third; and then the resulting following part of the jobs constituting r_1 *, l_k, $\cdots$, l_m; and after jobs in the same order as S_β.

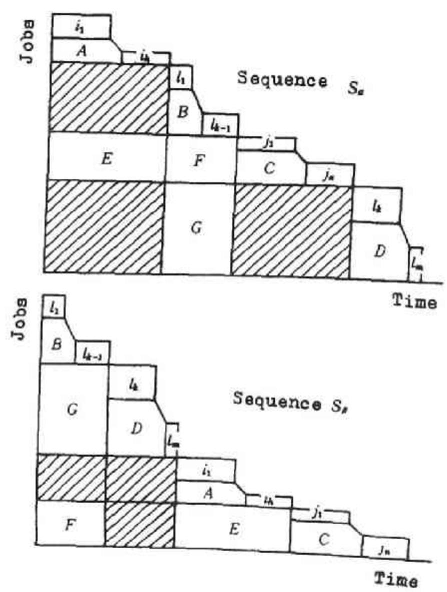


Fig. 2.6. Comparison of total weighted flow-times of S_α and S_β

Though this shows only the case in which the jobs constituting r_i^* , are divided into two parts, the same argument can be applied to any case however the jobs are divided. Therefore the case as in Fig. 2.6 is regarded as the proof of the general case without loss of generality. The different parts of sequences S_α and S_β are represented graphically in the manner of Fig. 2.6. The total area of this graph - both labeled blocks and shaded portion - represents the sum of the job flow-times of the different part. Notice that the area represented by the labeled blocks A, B, ... is common to S_α and S_β .

The total weighted flow-time of S_α - the total weighted flow-time of S_β $= (p_{i_1}+\cdots+p_{ih})(u_{1_1}+\cdots+u_{1_m})+(p_{j_1}+\cdots+p_{jn})(u_{1_k}+\cdots+u_{1m})$

$$-(p_{1_1}+\cdots+p_{1m})(u_{i_1}+\cdots+u_{i_h})-(p_{1_k}+\cdots+p_{1_m})(u_{j_1}+\cdots+u_{j_n})$$

$$= (u_{1_1}+\cdots+u_{1_m})(u_{i_1}+\cdots+u_{i_h})\left[\frac{p_{i_1}+\cdots+p_{i_h}}{u_{i_1}+\cdots+u_{i_h}}-\frac{p_{1_1}+\cdots+p_{1_m}}{u_{1_1}+\cdots+u_{1_m}}\right]$$

$$+(u_{1_k}+\cdots+u_{1m})(u_{j_1}+\cdots+u_{j_k})\left[\frac{p_{j_1}+\cdots+p_{jn}}{u_{j_1}+\cdots+u_{jn}}-\frac{p_{1k}+\cdots+p_{1m}}{u_{1_k}+\cdots+u_{1_m}}\right]$$

[By Theorem 2.2]

$$\geq (u_{1_1}+\cdots \text{i}\ u_{1_m})(u_{i_1}+\cdots+u_{i_h})\left[\frac{p_{i_1}+\cdots+p_{ih}}{u_{i_1}+\cdots+u_{ih}}-\frac{p_{1_1}+\cdots+p_{1_m}}{u_{1_1}+\cdots+u_{1_m}}\right]$$

$$+(u_{1_k}+\cdots+u_{1_m})(u_{j_1}+\cdots+u_{jk})\left[\frac{p_{j_1}+\cdots+p_{j_n}}{u_{j_1}+\cdots+u_{j_n}}-\frac{p_{1_1}+\cdots+p_{1m}}{u_{1_1}+\cdots+u_{1_m}}\right]$$

$$=(>0)(>0)[\geqslant 0]+(>0)(>0)[\geqslant 0]\geqslant 0$$

The first term [ ] of the right hand side of the inequality is easily shown to be non-negative, but it needs an explanation to show that the second term [ ] is non-negative. It is to be shown that if $a, c, d \geqslant 0$; $b, d, f > 0$; and $a/b < c/d$, $a/b \leq e/f$, then $a/b \leq (c+e)/(d+f)$, where,

$a = p_{1_1}+\cdots+p_{1_m}$, $b = u_{11}+\cdots+u_{1m}$, $c = p_{i_1}+\cdots+p_{i_h}$,

$d = u_{i_1}+\cdots+u_{i_h}$, $e = p_{j_1}+\cdots+p_{j_n}$, $f = u_{j_1}+\cdots+u_{j_n}$.

Let $a/b = k_1(c/d) = k_2(e/f)$, then $0< k_1 \leqslant 1$, $0< k_2 \leqslant 1$,

and $\dfrac{a}{b} = \dfrac{k_1 c+k_2 e}{d+f} = \dfrac{c+e}{d+f} - \dfrac{(1-k_1)c+(1-k_2)e}{d+f} \leq \dfrac{c+e}{d+f}$.

Therefore, the right hand side of the above inequality is non-negative. This indicates that S_α dominates S_β , and therefore it can be concluded that it is necessary to perform the set of jobs constituting r_1*, first to minimize the total weighted flow-time. With the same argument r_2*, r_3*, ... should be performed in this order.

It has been shown by Theorem 2.3 that r_i* plays an important role under precedence ordering restrictions. Actually Theorem 2.3 is applicable to some more general case in which a set of jobs can be divided into mutually exclusive independent subsets of jobs. In Fig. 2.1, for instance, jobs 4 and 9 are break points and so jobs 1,2, and 3, or jobs 5, 6, 7, and 8 are considered as mutually exclusive independent subsets to which Theorem 2.3 is applied. But jobs

10, 11, 12, 13, and 14 are hardly divided into mutually exclusive independent subsets and therefore Theorem 2.3 is not applicable. One way to overcome the difficulty is to manage somehow to segment the set of jobs into mutually exclusive independent subsets. By adding a precedence relationship $12 < 13$, or $13 < 12$ as shown in Fig. 2.7 to the original precedence diagram, the jobs 10, 11, 12, 13, and 14 can be divided into mutually exclusive independent subsets. Notice that no sequence is lost by this operation. The optimum solution is selected by taking the better one from the optimum solutions obtained from applying Theorem 2.3 to the new two precedence diagrams.

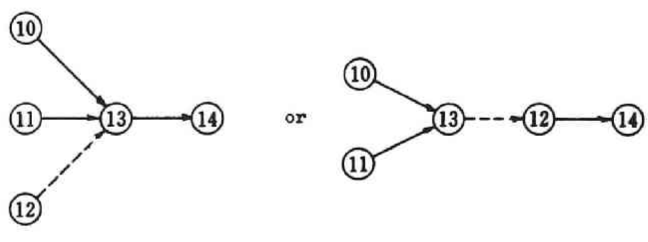


Fig. 2.7. Diving the jobs 10, 11, 12, 13, and 14 into two cases.

The same argument as Theorem 2.3 is applicable to the precedence diagram of tree-type as shown in Fig. 2.3. It matters little to calculate r_i* for small scale problems, but it requires a large amount of calculation for large scale problems. It is hoped that this will economize effort for calculation. Fortunately, the following easier way of calculation is found in this case : Parallel jobs following each job are rewritten in series in such a way that the ratio $r^*_{\text{each job}}$ is minimized, starting from the jobs of the last rank to the jobs of the earlier rank step by step, as shown in Fig. 2.8 (a), (b), and (c). By this operation effort for calculation is greatly reduced.

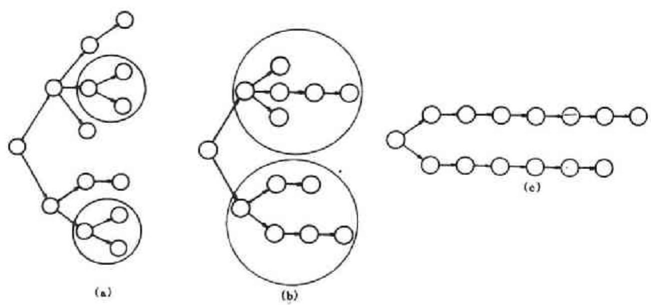


Fig. 2.8. Rewriting precedence relationships of tree type into series in such a way that $r^*_{\text{each job}}$ is minimized.

From the above analysis, Theorem 2.3 plays an essential role in the case when a set of jobs can be divided into mutually exclusive independent subsets of jobs without violating required precedence relationships. Theorem 2.3 reduce to the theorem introduced by Smith in case precedence ordering restrictions are not imposed.

Unfortunately, Theorem 2.3 is not applicable to more complex precedence diagrams, for example, as in Fig. 2.9. In Fig. 2.9, the numbers inside the circles identify jobs, and the number outside the circles give the processing-times. The relative importance of each job is assumed to be 1 .

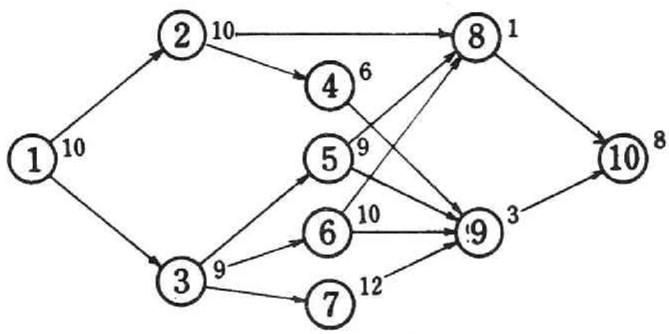


Fig. 2.9. An example to which Theorem 2.4 is not applicable.

Gapp et al[3] insist that the following theorem is effective in the case where precedence relationships are imposed .

<u>Theorem 2.4</u> If $i \leq j$ is imposed and $p_i/u_i > p_j/u_j$, then the optimal feasible sequence will contain i and j as an ordered adjacent pair (i, j) unless there exists an operation (or equivalent operation) k such that either

(a) $i < k$ and $k < j$, or

(b) $k < j$ and $p_i/u_i < p_k/u_k$, or

(c) $i < k$ and $p_j/u_j \geq p_k/u_k$.

(Proof) Omitted. See reference (3).

Theorem 2.4 is, however, hardly applied to the precedence diagram shown in Fig. 2.9. As far as job 1 is concerned there is no problem since it is performed first anyway. The next job or set of jobs to be processed are 2 [10], 3 [9], (2·4) [8], (3·5) [9], (3·6) [9.5], (3·7) [10.5], (3·5·6) [9.3], (3·5·7) [10], (3·6·7) [10.3], (3·5·6·7) [10]. Square bracket [ ] after each job or set of jobs shows r_i . After serious thought the possibility that the optimum solution can be obtained from selecting the job or the set of jobs which gives the minimum value among these r_i ' s will be considered. To discuss the possibility generally, let

- $F\{i\} \quad := \{j \mid i \ll j \ ; \ i, j \in X\}$.
- $F'\{i\} \quad := \{j \mid j = i \text{ or } i \ll j \ ; \ i, j \in X\}$.
- $F_A\{i\} \quad := \{j \mid i < j \ ; \ i, j \in X\}$.
- $F_A'\{i\} \quad := \{j \mid j = i, \text{ or } i < j \ ; \ i, j \in X\}$.
- $F\{i, j, \cdots, k\} \quad := \{l \mid l \ll m \ ; \ l, m \in F\{i\} + F\{j\} + \cdots + F\{k\}\}$.
- $F_A\{i, j, \cdots, k\} := \{l \mid l \quad F_A\{i\} + F_A\{j\} + \cdots + F_A\{k\}\}$.
- $F_A'\{i/j, \cdots, k\} := F_A'\{i\} - F_A\{j\} - \cdots - F_A\{k\}$. The set of jobs which include job i and jobs following i, and do not include the jobs following the jobs j, ... , k.

- $r^*_{i/j,\cdots,k}$: $\underset{\delta_\nu(i) \in F_A'\{i/j,\cdots,k\}}{\mathrm{Min}} \left[\sum_{i \in \delta_\nu(i)} p_i \Big/ \sum_{i \in \delta_\nu(i)} u_i \right]$. For instance,

  for jobs 2 and 3 in Fig.2.9, $F_A'\{2/3\} = (2,4)$, $\delta_1(2) = 2$,

  $\delta_2(2) = (2,4)$ and $r^*_{2/3} = \mathrm{Min}\left[p_2/u_2, \ (p_2 + p_4)/(u_2 + u_4) \right] = 8$.

- $\lambda\{r_{i\nu}\}$: The set of jobs constituting $r_{i\nu}$.
- X(i) : The job or the set of jobs assigned in the i th position of an obtained sequence.

The problem is to discuss the possibility whether an optimum sequence can be obtained from selecting the job or the set of jobs which gives the minimum value among $r^*_{i_1/i_2,\cdots,i_m}$, $r^*_{i_2/i_1,\cdots,i_m}$, ..., $r^*_{i_m/i_1,i_2,\cdots,i_{m-1}}$, given an optimum partial sequence $\{X(1), X(2), \cdots, X(i-1)\}$ and that jobs $i_1, i_2, \ldots, i_m$ can be processed next. In what follows this is proved subject to certain conditions.

<u>Theorem 2.5.</u> Suppose that an optimum partial sequence $\{X(1), X(2), \cdots, X(i-1)\}$ has been obtained already, and that jobs $i_1, i_2, \ldots, i_m$ can be performed next. After deleting the jobs included in $\{X(i), X(2), \cdots, X(i-1)\}$ from the original precedence diagram, if the following conditions :

(i) $F\{\lambda(r^*_{i_1/i_2,\cdots,i_m})\} \in F_A\{\lambda(r^*_{i_2/i_1,\cdots,i_m})\}$

(ii) $r^*_{i_2/i_1,\cdots,i_m} \leq r^*_{i_1/i_2,\cdots,i_m}$

are satisfied, $\lambda(r^*_{i_2/i_1,\cdots,i_m})$ dominates $\lambda(r^*_{i_1/i_2,\cdots,i_m})$

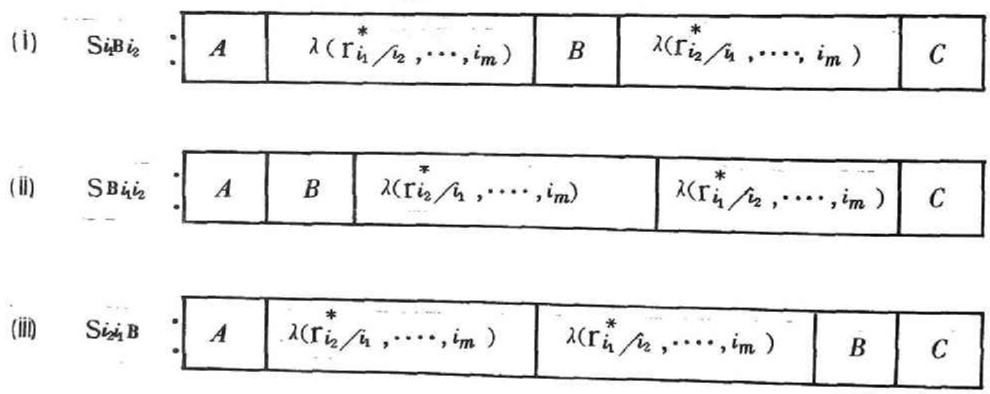


Fig. 2.10. Proof of Theorem 2.5.

(Proof) The assumption (i) means that the precedence relationships which $\lambda(r^*_{i_2}/i_1,\cdots,i_m)$ has are tighter than those of $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$, that is, positions of $\lambda(r_{i_1}/i_2,\cdots,i_m)$ and $\lambda(r_{i_2}/i_1,\cdots,i_m)$ can be interchanged without disturbing precedence relationships in any sequence in which $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$ precedes $\lambda(r^*_{i_2}/i_1,\cdots,i_m)$, but the converse is not always possible. Let $S_{i_1 B i_2}$ denote the best sequence in which $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$ precedes $\lambda(r^*_{i_2}/i_1,\cdots,i_m)$ as shown in Fig. 2.10.(i). Also let A, B, and C denote the jobs which precede $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$, the jobs which are between $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$ and $\lambda(r^*_{i_2}/i_1,\cdots,i_m)$, and the jobs which follow $\lambda(r^*_{i_2}/i_1,\cdots,i_m)$, respectively. As shown in Fig. 2.10.(ii), and (iii), let $S_{B i_2 i_1}$, and $S_{i_2 i_1 B}$ be the sequence which processes A, B $\lambda(r^*_{i_2}/i_1,\cdots,i_m)$, $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$ and C in this order, and the sequence which performs A, $\lambda(r^*_{i_2}/i_1,\cdots,i_m)$, $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$, B and C in this order, respectively, then these two sequences $S_{B i_2 i_1}$ and $S_{i_2 i_1 B}$ are also feasible. Let p_B, p_{λ_1}, denote the total processing-times of B, $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$ and u_B, u_{λ_1}, the total processing-times of B, $\lambda(r^*_{i_1}/i_2,\cdots,i_m)$, etc.

(i) If $p_B/u_B \le p_{\lambda_2}/u_{\lambda_2} = r^*_{i_2}/i_1,\cdots,i_m$, then

$$p_B/u_B \le (p_B+p_{\lambda_2})/(u_B+u_{\lambda_2}) \le p_{\lambda_2}/u_{\lambda_2},$$

therefore,

the total weighted flow-time of $S\,i_1\,B\,i_2$ - the total weighted flow-time of $S\,B\,i_1\,i_2$

$$= u_{\lambda_1}\,(u_B + u_{\lambda_2})\left[\frac{p_{\lambda_1}}{u_{\lambda_1}} - \frac{p_B + p_{\lambda_2}}{u_B + u_{\lambda_2}}\right] \geqslant 0 \quad .$$

(ii) If $p_B/u_B > p_{\lambda_2}/u_{\lambda_2}$, then

the total weighted flow-time of $S\,i_1\,B\,i_2$ - the total weighted flow-time of $S\,B\,i_1\,i_2$

$$= (u_{\lambda_1} + u_B)\,u_{\lambda_2}\left[\frac{p_B + p_{\lambda_1}}{u_B + u_{\lambda_1}} - \frac{p_{\lambda_2}}{u_{\lambda_2}}\right] \geqslant 0 \;.$$

For the sequence as shown in Fig. 2.10.(i), a better sequence as in either Fig. 2.10(ii), or (iii), can always be found and therefore it can be concluded that $\lambda\,(r^*_{i_2}/i_1, \cdots, i_m)$ dominates $\lambda\,(r^*_{i_1}/i_2, \cdots, i_m)$.

An application of Theorem 2.5 to the precedence diagram shown in Fig. 2.9 results in the optimum solution

$\pi = 1\cdot 2\cdot 4\cdot 3\cdot 5\cdot 6\cdot 8\cdot 7\cdot 9\cdot 10$ [The total weighted flow-time is 459].

Notice that the assumption (i) in Theorem 2.5 is essential. The job which can be performed next among 10, 11, 12, 13 and 14 in Fig. 2.1 is either 10, or 11, or 12. The relative importance of each job is assumed to be 1. The optimum sub-sequence of jobs 10, 11, 12, and 13 is $\pi = 10\cdot 11\cdot 13\cdot 12$

although job 12 has the smallest processing-time. This is due to the fact that the assumption (i) is not satisfied in this case, which can be solved by the operation of rewriting the precedence diagram. To think it generally, it is considered why the set of jobs which gives the minimum ratio among

$r^*_{i_1}/_{i_2,\cdots,i_m},\ r^*_{i_2}/_{i_1,\cdots,i_m},\ \cdots,\ r^*_{i_m}/_{i_1,i_2,\cdots,i_{m-1}},$

can not always be chosen given that $i_1, i_2, \ldots,$ and i_m can be processed next. Among job 10, 11, 12, 13, and 14 in Fig. 2.1, the set of jobs (11, 13) can be performed after job 10 has been processed, and $r_{11}^* = 2.1 < r_{12}^* = 3$. This fact prevents processing job 12 earlier from being optimum. It is well understood since the optimality in Theorem 2.3 has been dependent on the non-decreasing property of r_i^*. From this discussion, the following theorem will be established :

<u>Theorem 2.6.</u> Under the same assumptions of the first half of Theorem 2.5, if

$r^*_{i_1}/_{i_2,\cdots,i_m} \leq \mathrm{Min}[r^*_{i_2}/_{i_1,\cdots,i_m},\ r^*_{i_3}/_{i_1,\cdots,i_m},\ \cdots,\ r^*_{i_m}/_{i_1,\cdots,i_{m-1}},\ r^*_{i_2},\ r^*_{i_3},\ \cdots,\ r^*_{i_m}]$

then $\lambda(r^*_{i_1}/_{i_2,\cdots,i_m})$ should be performed next.

(Proof) Let S_{i_1} be the optimum sequence obtained by performing $\lambda(r^*_{i_1}/_{i_2,\cdots,i_m})$ next. Also let $\lambda(r_j/_{i_1,\cdots,i_m,\ \mathrm{except}\ j})$ [where $j \in \{i_2,\cdots,i_m\}$] represent the jobs which follow $\lambda(r^*_{i_1}/_{i_2,\cdots,i_m})$ in sequence S_{i_1} and which can be performed before $\lambda(r^*_{i_1}/_{i_2,\cdots,i_m})$ by interchanging positions without disturbing precedence relationships. Let S_j be the sequence in which $\lambda(r_{j}/i_1, \ldots, i_m, \text{except } j)$ precedes $\lambda(r^*_{i_1}/i_2, \ldots, i_m)$. Then, from the assumption, it is easily shown that the total weighted flow-time of S_j - the total weighted flow-time of $S_{i_1} \geqslant 0$.

Suppose, for instance, that the processing-time of job 13 in Fig. 2.1 is 3 instead of 1, then, Theorem 2.6 insists on processing job 12 first among job 10, 11, and 12. However, if the processing-time of job 13 is 1 as it was before, Theorem 2.6 does not insist on processing job 12 first, nor give further information. In this case, it should be decided whether job 12 should be processed first or not according to the sign of the following expression :

$$u_{10} \cdot u_{12}(r_{10} - r_{12}) + (u_{11} + u_{13}) \cdot u_{12} \cdot (r_{11,13} - r_{12}) \quad (\gtreqless 0) \ .$$

A new theorem could be established by considering the case generally, but as is easily understood, checking whether assumptions of theorems are satisfied or not for such cases would become quite troublesome for large scale problems. From this point of view, it is surmised that Theorem 2.1~2.6 should be used as far as they can apply and that if they are not applicable, given precedence relationships should be rewritten by adding more precedence constraints.

2.2.4 The Algorithm

The algorithm is now developed.

Algorithm.

Step 1-A. For the given precedence diagram, pick up the job or construct the set of jobs which can be performed first.

Step 1-B. Reduce the existential range of an optimum solutio by applying Theorem 2.1 ~ 2.6 to the collection obtained in Step 1-A.

Step 1-C. Write List 1, the list of partial sequences $\{X(1)\}$ of a set of jobs, with $X(1)$ in the collection obtained in Step 1-B.

Step i-A, $i \geqslant 2$. For each sequence $\{X(1), \cdots, X(i-1)\}$ of sets of jobs on List (n-1), pick up the job or construct the set of jobs which can be performed next after $\{X(1), \cdots, X(i-1)\}$.

Step i-B. Delete jobs or sets of jobs dominated by other jobs or sets of jobs from the collection obtained in Step i-A.

Step i-C. Write List i-C, the list of sequences $\{X(1), \cdots, X(i)\}$ with $\{X(1), \cdots, X(i-1)\}$ on List (i-1), and $X(i)$ in the collection of next assignments after $\{X(1), \cdots, X(i-1)\}$ . .

Step i-D. Obtain List i from List i-C, by successively crossing off redundant sequences for which there is another sequence $\{X(1), \cdots, X(i)\}$ on the list (still not crossed off) ; such that each job included in a redundant sequence is included in some $X(j)$ and that the total weighted flow-time of the sequence is greater than or equal to that of $\{X(1), \cdots, X(i)\}$.

Repeat the steps as many times as necessary.

Notice that rewriting given precedence relationships corresponds to listing possible sequences.

The optimum sequences for the precedence diagrams shown in Fig. 2.1 and Fig. 2.9 are obtained quite easily by the algorithm, but for large scale problems the following heuristic algorithm should be used.

The heuristic algorithm.

Step 1-A. For the given precedence diagram, pick up the job or construct the set of jobs which can be performed first.

Step 1-B. Calculate $r^*_{i/j, \cdots, k}$ for each job, or set of jobs obtained in Step 1-A and select the job, or set of jobs which gives the minimum ratio among $r^*_{i/j, \cdots, k}$.

Step i-A, $i \geqslant 2$. Cross off from the given precedence diagram the jobs included in the selected partial sequence obtained in Step (i-1), and pick up the job or construct the set of jobs which can be performed next.

Step i-B. Calculate $r^*_{i/j, \cdots, k}$ for each job, or set of jobs obtained in Step i-A and select the job, or set of jobs which gives the minimum ratio among $r^*_{i/j, \cdots, k}$.

Repeat the steps as many times as necessary.

This heuristic algorithm, of course, does not always give optimum solutions.

2.2.5 Discussions

By developing several useful theorems, it has been made clear that when precedence relations are imposed, it is more essential and important to consider

$$\Sigma \text{ (processing time) } / \ \Sigma \text{ (weighted coefficient)}$$

of the jobs related to each other instead of

$$\text{(processing time) } / \text{ (weighted coefficient)}$$

of each job, and that there is still limitation in applying the above value to an arbitrary precedence diagram. Instead of establishing more theorems by introducing complicated assumptions requiring time and pains for examining if the assumptions are satisfied, a method has been introduced by which an arbitrary precedence diagram is changed into independent precedence relations to which the above value can be applied, and an effective new algorithm to minimize mean weighted flow time with precedence restrictions has been developed.

2. 3 SEQUENCING A SET OF JOBS TO MINIMIZE THE SUM OF SEQUENCE-DEPENDENT SETUP-TIMES WITH PRECEDENCE RESTRICTIONS

2.3.1 Problem Statement

This section considers the case in which the setup-time is sequence dependent — and the jobs have required precedence relationships. In fact, there are many situations in which the variation of setup-time with sequence provides the dominated criterion for evaluating a sequence. Suppose there is a facility which operates on one product at a time, but on distinctly different products in sequence. Certain jobs can have similar setups so that changing from one to another is simply a matter of adjusting stops and perhaps changing tools. The other jobs on the same facility could require an entirely different setup. Let the jobs be indexed by $i = 1, 2, \dots, n$. The precedence relationships among the jobs are displayed on a precedence diagram and are represented analytically by the fundamental matrix L_0. Notation $s(i,j)$ represents the times to change over from job i to job j. Setup-times between all possible job pairs are presumed known. When the jobs have required precedence relationships, $s(i,j)$ is given to an ordered pair whose transition is possible. These $s(i,j)$ form a matrix. The matrix is generally nonsymmetric, for the time to change over from job i to job j is, in general, different from the time to change over from job j to job i. In what order should the jobs be performed to minimize the total setup-times consumed without breaking required precedence relationships. A solution to the problem, i.e., a linear sequence is given by a set of (n-1) ordered job pairs, e.g.,

$$\pi = (1,2)\ (2,3)\ \ldots\ (n-1,n)$$

The value of a solution π, is the sum of the matrix elements picked out by π and will be denoted by $z(s)$:

$$z(\pi) = \sum_{(i,j)\in\pi} s(i,j)\ .$$

Sequence π always picks out one and only one value in each column and in each row without breaking required precedence relationships.

The problem reduces to the traveling salesman problem if it does not have required precedence relationships. The traveling salesman problem is noted for its difficulty which is entirely computational, since a solution obviously exists. In recent years a number of methods for solving the traveling salesman problem have been put forward[5)~17)].

The problem is more difficult than the traveling salesman problem since it has required precedence relationships in addition to the difficulty of the traveling salesman problem.

In case of the traveling salesman problem which has no required precedence relationships there are (n-1)! possible sequences, assuming that the beginning job (i.e. city) is fixed, while in the problem tackled in this section, only some of these (n-1)! possible sequences will be feasible because of required precedence relationships, as discussed already in Chapter 1. This fact should be used to reduce computational efforts to find one or more optimum sequences which must give minimum time. Based on the fact, two effective algorithms, computationally feasible, will be formulated in the following.

2.3.2 The Two Algorithms

The algorithms are based on the Branch and Bound approach. The practical success of applying a branch and bound approach to solve an actual combinatorial problem depends considerably on exploiting the special structure of its model.

The basic idea of the algorithms developed in the following is to break up the set of all linear sequences into smaller and smaller subsets without disturbing required precedence relationships and to calculate for each of them a lower bound on the time of the best linear sequence therein. Two main problems are how to break up the set of all linear sequences without violating the precedence relationships and how to get for each subset of linear sequences a good lower bound which is hopefully close to the time that the best linear sequence in that subset has. Two algorithms will be developed in the following, to one of which the consideration of getting a good lower bound is given more weight than that of not disturbing required precedence relationships and to another, vs.

The algorithm will simultaneously be explained and illustrated by a numerical example. It is shown in Fig. 2. 11.(a). The fundamental matrix L_0 with setup-times is given in Fig. 2. 11. (b). The numbers written in quarter-circles made at the southeastern corners of the boxes of rows and columns show the setup-times.

(1) Algorithm 1.

In this algorithm the consideration of not violating required precedence relationships is given more weight than that of getting a good lower bound. The characteristics of the algorithm are as follows :

(i) Sequential multiplication applies correspondingly to the process of branching.

(ii) Obtaining lower bounds is based on the assignment model optimization.

It is necessary to employ the notation of a partial sequence, which was defined in Chapter 1 as a sequence of less than n distinct jobs, starting with job 1. Extending a partial sequence from job i, which is the last job in the partial sequence, to job j is subject to sequential product.

The process of branching which splits the set of all feasible complete linear sequences into disjoint subsets which indicate partial sequences will be represented by a tree-like diagram as illustrated in Fig. 2. 12 : The knot Γ_1 represents the set of all feasible linear sequences. Consider how much time is necessary to complete all of the jobs without violating given precedence relationaships. The time, which is called lower bound on time, may be obtained by reducing rows and columns in L_0 . The process of subtracting the smallest element of a row (column) from each element in the row (column) will be called reducing the row (column). By this operation, each row and each column in L_0 has non-negative elements and at least one zero as far as setup-times are concerned. This transformed matrix is called the reduced matrix of L_0 and denoted by L_1 . In what follows, reduced matrices which give only setup-times for the sake of convenience are shown by notation L with its suffix. The reduced matrix for Fig. 2. 11 is shown in Fig. 2. 13. However, maximum independent 0 's can not always be obtained

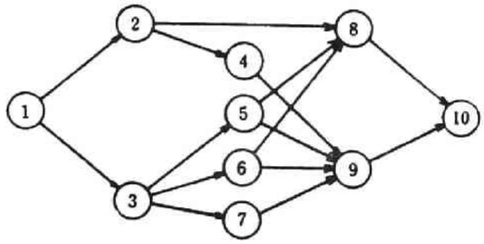


Fig.2.11. (a) Precedence diagram.

$L_0 =$

	1	2	3	4	5	6	7	8	9	10
1	0	2_{11}	$3_{[illegible]}$	0	0	0	0	0	0	0
2	0	0	3_{55}	4_{6}	$^{3}5_{9}$	$^{3}6_{11}$	$^{3}7_{3}$	$^{3,6}8_{82}$	0	0
3	0	2_{6}	0	$^{2}4_{97}$	5_{85}	6_{94}	7_{24}	0	0	0
4	0	0	3_{20}	0	$^{3}5_{25}$	$^{3}6_{24}$	$^{3}7_{[illegible]}$	$^{5,6}8_{74}$	$^{5,6,7}9_{75}$	0
5	0	2_{81}	0	$^{2}4_{15}$	0	6_{19}	7_{74}	$^{2,6}8_{87}$	$^{4,6,7}9_{23}$	0
6	0	2_{81}	0	$^{2}4_{85}$	5_{35}	0	$7_{[illegible]}$	$^{2,5}8_{29}$	$^{4,5,7}9_{68}$	0
7	0	2_{68}	0	$^{2}4_{82}$	5_{34}	6_{8}	0	$^{2,5,6}8_{85}$	$^{4,5,6}9_{62}$	0
8	0	0	0	4_{42}	0	0	7_{7}	0	$^{5,7}9_{12}$	$^{9}10_{65}$
9	0	0	0	0	0	0	0	8_{81}	0	$^{8}10_{2}$
10	0	0	0	0	0	0	0	0	0	0

(b) Fundamental matrix with setup-times.

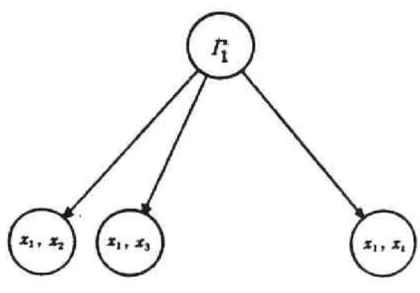


Fig.2.12. Illustration of a tree-like diagram.

by this operation, as is easily understood. It means that it might be possible to have a higher lower bound. It is hopeful to have as high a lower bound as possible so that the algorithm will converges quickly, disregarding calculation time. By the assignment model optimization maximum independent 0's can be obtained. Fig. 2. 14 is obtained by applying the assignment model optimization to Fig. 2. 13. The transformed matrix is denoted by L_i with ' ' ' . The knot Γ_1 has a lower bound $l(\Gamma_1)$ on the optimal value of the objective function. In the example, summing up the reduced constants used for reducing the matrix and for obtaining matrix independent 0's by the assignment model optimization, results in

$$l(\Gamma_1) = 11+5+6+20+15+29+8+7+2+4+5+18 = 130$$

The next operation is to check by sequential product whether or not the solution of the assignment model optimization satisfies the required precedence relationships. If it satisfies them, it surely is an optimum solution.

	1	2	3	4	5	6	7	8	9	10
1		0	77							
2			48	1	0	6	0	87		
3		0		91	23	88	18			
4			0		4	4	69	54	50	
5		46		0		4	59	52	3	
6		52		54	0		69	0	34	
7		60		74	22	0		75	49	
8				35			0		0	56
9								59		0
10										

Fig.2.13. Reduced matrix of L_0 : L_1 .

	1	2	3	4	5	6	7	8	9	10
1		(0)	59							
2			48	1	(0)	6	0	87		
3		0		73	5	70	(0)			
4			(0)		4	4	69	54	50	
5		64		(0)		4	59	52	3	
6		70		54	0		69	(0)	34	
7		78		74	22	(0)		75	49	
8				35			0		(0)	56
9								59		(0)
10										

$\Pi = 1 \cdot 2 \cdot 5 \cdot 4 \cdot 3 \cdot 6 \cdot 8 \cdot 9 \cdot 10$

Fig.2.14. Transformed matrix by the assignment model optimization : L_1' .

In the example, the solution is

$$\pi = 1.2.3.4.5.6.7.8.9.10 \ .$$

This π is not feasible. Therefore, it is necessary to make branching from knot 'Γ_1'. Knot '(x_1, x_i)' represents the set of all complete linear sequences composed from a partial sequence from x_1 to x_j. The number of the knots branched from 'Γ_1' is equal to that of the jobs capable of direct transition from x_1. The cardinal number of the set represented by '(x_1, x_i)' denotes the total number of all complete linear sequences produced from partial sequence $\pi(x_1, x_i)$. Therefore, the sum of the cardinal numbers of the sets represented by the knots after branching is equal to the cardinal number of the knot before branching. This means there is no sequence lost by branching. At any stage of the process of branching, the union of the sets represented by the terminal knots, which are defined as the unbranched knot, is the set of all feasible complete linear sequences.

Suppose that the direct transition from x_1 to x_i is possible. Then, the lower bound $l\{(x_1, x_i)\}$, which knot '(x_1, x_i)' will have, gets an increase of the time from x_1 to x_i in matrix L_1', i.e., $S_{L_1'}(x_1, x_i)$, in addition to $l(\Gamma_1)$. Since the job pair (x_1, x_i) is now committed to the sequences, row x_1 and column x_i are no longer needed and are deleted from L_1'. If $c(i,p)$ in row x_i and in column p does not satisfy the following condition :

$$c(i,p) \in S\{(x_1, x_i)\}$$

then the transition from x_1 to x_p should be forbidden for it would create an infeasible sequence. Therefore, set

$$s_{L_1'}(i,p) = \infty$$

The resulting (n-1, n-1) matrix is transformed to the reduced matrix $L_2(x_i)$, and then by the assignment model optimization $L_2'(x_i)$. The sum of reducing constants, $R_{L_1'}(x_1,x_i)$, used for reducing the matrix and for obtaining maximum independent O's by the assignment model optimization also increases the lower bound. Therefore,

$$l\{(x_1, x_i)\} = l(\Gamma_1) + s_{L_1}(x_1, x_i) + R_{L_1}(x_1, x_i)$$

In general there are more than one jobs which are reached by direct transition from x_1 . The lower bound for each of them must be calculated in the same manner. The partial sequence (x_1, x_i) which has the minimum lower bound among these is selected. There is no guarantee of the optimality of the complete linear sequence whose part is a partial sequence (x_1, x_i), but, at this step, (x_1, x_i) has the highest possibility of going to the optimum.

For the example, the jobs which are reached by direct transition from 1 are 2 and 3. Deleting row 1 and column 2, or row 1 and column 3, from L_1' and applying the assignment model optimization to them result in $L_{2,2'}$, $L_{2,3'}$ respectively as shown in Fig. 2.15. Notice that the times in row 2 and in column 5,6,7 and 8 in $L_{2,2'}$, and the time in row 3 and in column 4 in $L_{2,3'}$ are set to ∞ .

$$l\ (1,2) = 157$$

$$l\ (1,3) = 193$$

The solution of the assignment model optimization in the first case is $\pi = (1\cdot 2\cdot)4\cdot 3\cdot 7\cdot 5\cdot 6\cdot 8\cdot 9\cdot 10$

$L_{2,2'}$

	1	3	4	5	6	7	8	9	10
2		21	(0)						
3			99	5	92	(0)			
4		(0)		4	26	69	54	73	
5			0		(0)	33	26	0	
6			80	0		69	(0)	57	
7			78	(0)	0		53	50	
8			38					(0)	33
9						0	59		10
10									

$L_{3,3'}$

	1	2	4	5	6	7	8	9	10
2			1	0	6	(0)	87		
3		(0)		23	88	18			
4				(0)	0	65	50	46	
5		46	(0)		4	59	52	3	
6		52	54	0		69	(0)	34	
7		60	74	22	(0)		75	49	
8			35			0		(0)	56
9							59		(0)
10									

Fig.2.15. $L_{2,2'}$ and $L_{2,3'}$

Checking the solution by sequential product makes it clear that it is feasible and therefore is optimal. The tree-like diagram is shown in Fig. 2.16. For reference, the tree-like diagram of the case in which the assignment model optimization is not used is shown in Fig. 2.17.

From the above discussion, an algorithm can be developed.

Algorithm 1

Step 1-A. Construct the fundamental matrix L_0 for the given set of jobs $X = \{1, 2, \ldots, n\}$. Make knot 'Γ_1' in a tree-like diagram.

Step 1-B. Reduce L_0 with respect to the setup-times and obtain the reduced matrix L_1. Apply the assignment model optimization to L_1 and obtain L_1'. Obtain the first solution, and calculate the lower bound, $l(\Gamma_1)$. Give the value to knot 'Γ_1'.

Step 1-C. Check by sequential product whether or not the solution satisfies the required precedence relationships. If feasible, stop, since the solution is an optimal feasible sequence, otherwise let $\pi = 1$ and go to Step m-A.

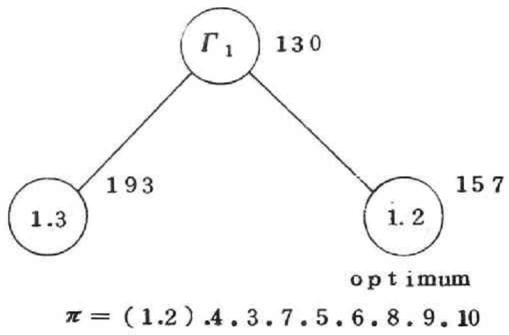


Fig.2.16. Tree-like diagram for the example.

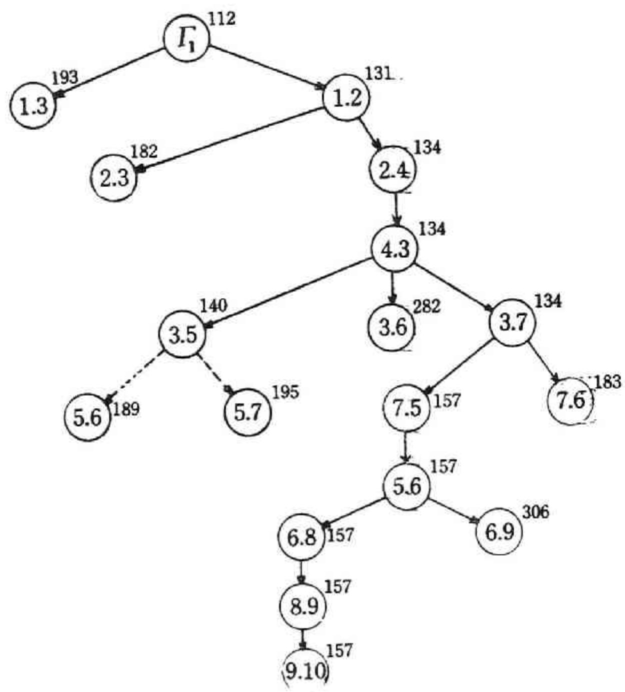


Fig.2.17. Tree-like diagram for the case in which the assignment model optimization is not used.

Step m-A, $m \geq 2$. Extend the selected partial sequence π , from job i, which is the last job in π , to jobs j, which are reached from i by direct transition. Make branches of knot '(i,j)'.

Step m-B. Perform for each of the jobs j the following (i), (ii), and (iii) :

(i) Cross off row i and column j from the matrix $L_{m-1,i}'$, where suffix (m-1) shows a matrix produced at step (m-1), and in addition to this, i should be added for distinction since generally more than one matrix are produced at step (m-1).

(ii) If for c(j,q) in row j and in column q,

$$c(j,q) \notin S\{\pi\}$$

then set $s_{L_{m-1},i'}(j,q)=\infty$

(iii) Reduce the matrix and obtain the reduced matrix $L_{m,j}$. Apply the assignment optimization model to $L_{m,j}$ and obtain $L_{m,j}'$. Obtain the solution, and calculate the lower bound,

$l\{\pi(x_1 \cdots x_i \cdot x_j)\}$:

$$l\{\pi(x_1 \cdots x_i \cdot x_j\} = l\{\pi(x_1 \cdots x_i)\} + s_{L_{m-1},i'}(x_i, x_j) + R_{L_{m-1},i}(x_i, x_j) \ .$$

Give the value to knot '(i,j)' .

(iv) Select the solution associated with the job pair (i,j) which gives the minimum lower bound.

Step m-C. Check by sequential product whether or not the solution satisfies the required precedence relationships.

(i) If feasible, terminate branching thereafter from knot '(i,j)'.

① If the lower bound of knot '(i,j)' does not exceed any lower bound of the unbranched knots, stop, since the solution is an optimal feasible sequence.

② Otherwise, select the unbranched knot which has the minimum lower bound.

Ⓐ If the knot is terminated, stop, the solution is an optimum feasible sequence ;

Ⓑ otherwise, go to step m-A.

(ii) If not feasible, go to step m-A.

If the steps are carried far enough, an optimum feasible sequence will eventually be produced.

(2) Algorithm 2 .

In this algorithm the value which Little et al[14] introduced is used :

$\theta(i,j)$ = the time of the smallest element in row i, excluding $s(i,j)$ + the time of the smallest element in column j, excluding $s(i,j)$.

In the tree-like diagram associated with Algorithm 2, a knot either branches into two further knots, or does not branch. The knot containing (i,j) represents all sequences which include the job pair (i,j), while the knot containing

$\overline{(i,j)}$ represents all sequences which do not.

In general, by tracing from a knot back to the start, it can be shown which job pairs are committed to appear in the sequences of the knot and which are forbidden from appearing. After branching according to the time, with the job pair having the smallest value being sequenced first, the lower bounds on the knot with the newly committed job pair and on the knot with the newly forbidden job pair, are calculated. If the solution associated with the newly committed job pair satisfies required precedence relationships, it is unnecessary to branch thereafter ; Otherwise it is necessary to branch further. To do it without producing infeasible sequences, some logical operation is required. For this the following fact is effectively used :

"If there exists an arrow from a node, say, x_p , to a node, say, x_q in a subgroup, a direct transition from node x_p to any node in other subgroups is not possible. Moreover a direct transition from any node in other subgroups to node x_q is not possible."

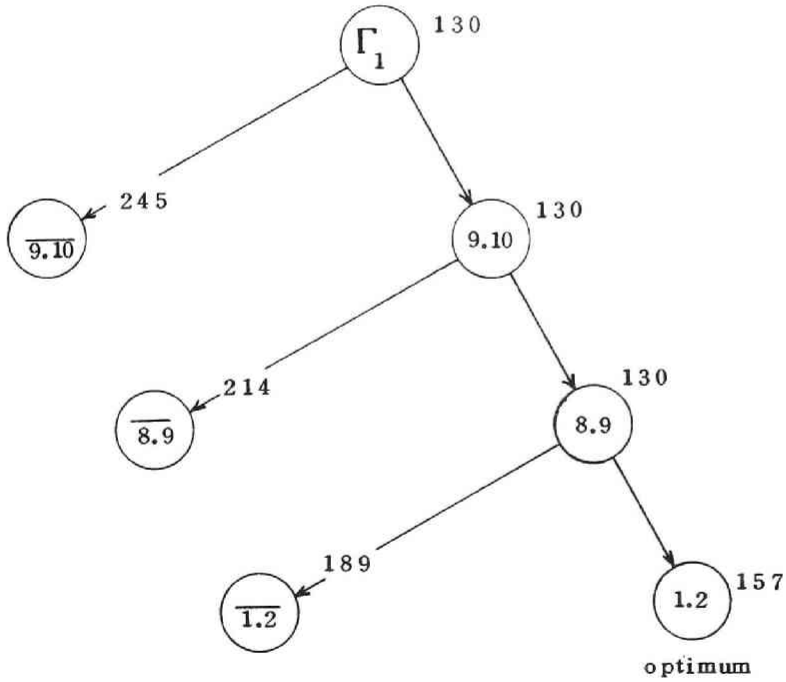
Either a job pair, or some of the job pairs which are committed to appear in the sequences of a knot may construct a subgroup. For the example, $\theta(i,j)$ are given in Fig. 2.18 for L_1' shown in Fig. 2.14 . According to the time, the job pair (9,10) is picked up, and two knots, (9,10) and $\overline{(9,10)}$

are branched from the start as shown in the upper part of Fig. 2.19. By selecting the job pair (9,10), the job pairs which are forbidden from appearing will come out.

	1	2	3	4	5	6	7	8	9	10
1		$(0)^{59}$	59							
2			48	1	$(0)^{0}$	6	0^{0}	87		
3		0^{0}		73	5	70	$(0)^{0}$			
4			$(0)^{52}$		4	4	69	54	50	
5		64		$(0)^{4}$		4	59	52	3	
6		70		54	0^{0}		69	$(0)^{52}$	34	
7		78		74	22	$(0)^{26}$		75	49	
8				35			0^{0}		$(0)^{38}$	56
9								59		$(0)^{115}$
10										

Fig.2.18. Calculation of $\theta(i,j)$ for L_1' .

They are found by investigating the directly preceding jobs of 9 and the directly following jobs of 10 under the fundamental matrix as mentioned above.



$\pi = (1.2).4.3.7.5.6.(8.9.10)$

Fig.2.19. Tree-like diagram by the algorithm 2 .

There is no job directly following 10. The directly preceding jobs of 9 are found to be 4, 5, 6, 7, and 8 from the fundamental matrix. Since there are two precedence relationships, $5 < 8$ and $6 < 8$ among the jobs 5, 6, and 8, the direct transitions from 5, and 6 to 9 become impossible. Changing the times from 5 to 9, and from 6 to 9, to ∞ and applying the assignment model optimization to the resultant matrix results in Fig. 2.20.

	1	2	3	4	5	6	7	8	9	10
1		$(0)^{59}$	59							
2			48	1	$(0)^{0}$	6	0^{0}	87		
3		0^{0}		73	5	70	$(0)^{0}$			
4			$(0)^{52}$		4	4	69	54	50	
5		64		$(0)^{5}$		4	59	52		
6		70		54	0^{0}			$(0)^{52}$		
7		78		74	22	$(0)^{26}$	0^{0}	75	49	
8				35					$(0)^{84}$	

Fig.2.20. Resultant matrix and $\theta(i,j)$. Note the values of (5,9) and (6,9) elements are set to ∞.

Since the solution by applying the assignment model optimization to Fig. 2.20 does not give a feasible sequence, more branching is required. Carrying branching far enough to get an optimum sequence, results in the tree-like diagram shown in Fig. 2.19. From the above discussion, another algorithm can be developed.

Algorithm 2

Step 1. The same as Step 1 of Algorithm 1.

Step m-A, $m > 2$.

(i) Calculate $\theta(i,j)$ for 0 elements in matrix L_{m-1}. Since only two matrices are produced at Step (m-1)-B, they are denoted by L_{m-1} and $\overline{L}_{m-1}$.

(ii) Pick up the job pair (i,j) which gives the minimum $\theta(i,j)$. Branch from Γ_{m-1}, $\Gamma_m=(i,j)$, and $\overline{\Gamma}_m=\overline{(i,j)}$.

(iii) Give the following lower bound to $\overline{\Gamma}_m=\overline{(i,j)}$:

$$l(\overline{\Gamma}_m)=l(\Gamma_{m-1})+\theta(i,j)$$

Step m-B.

(i) Cross off row i and column j from matrix L_{m-1}'.

(ii) Find the job pairs which are forbidden from appearing, and set the values associated with them to ∞.

(iii) Reduce the resultant matrix and obtain the reduced matrix L_m. Apply the assignment model optimization to L_m, and obtain L_m'. Obtain the solution and calculate the lower bound, $l(\Gamma_m)$:

$$l(\Gamma_m)=l(\Gamma_{m-1})+R_{L_{m-1}'}(i,j)$$

where, $R_{L_{m-1}'}(i,j)$ is the sum of reducing constants to obtain L_m' from L_{m-1}'. Give the value to knot '$l(\Gamma_m)$'.

Step m-C. Check by sequential product whether or not the solution satisfies required precedence relationships.

(i) If feasible, terminate branching thereafter from knot Γ_m'.

① If the lower bound of Γ_m' does not exceed any lower bound of the unbranched knots, stop, since the solution is an optimal feasible sequence ;

② Otherwise, select the unbranched knot which has the minimum lower bound.

(A) If the knot is terminated, stop, the solution is an optimum feasible sequence.

(B) Otherwise, let the knot be newly denoted by Γ_{m-1} , and go to Step m-B .

(ii) If not feasible, go to Step m-A.

Step m-D.

(i) Find the job pairs (i,j) which are committed to appear in the sequence of the knot, and calculate the sum of the times s(i,j), and cross off row i, and column j, associated with the job pairs (i,j) :

$$s = \sum_{(i,j)\in \text{ partial sequence of the knot}} s(i,j)$$

(ii) Find the job pairs which are forbidden from appearing in the sequences of the knot, and set the times associated with the job pairs to ∞.

(iii) Reduce the resultant matrix and obtain the reduced matrix L_m . Apply the assignment model optimization to L_m, and obtain L_m' . Calculate the lower bound $l(\Gamma_m)$:

$$l(\Gamma_m) = s + (\text{the sum of reducing constants to obtain } L_m')$$

Give the value to 'Γ_m'' . Obtain the solution, and go to Step m-C.

If the Steps are carried to a sufficient degree, an optimum feasible sequence will eventually be produced.

2.3.3 Discussions

The prcedence diagram shown in Fig. 2.11 has 246 feasible linear sequences. The steps needed to obtain an optimum feasible sequence are two, and four by Algorithms 1, and 2, respectively. If the problem had no precedence restrictions,there would be more than 3,600,000 permutation, and therefore probably much more steps would be required to obtain an optimum permutation. To solve the traveling salesman problem by the branch and bound algorithm requires an extensive memory. The two algorithms introduced in the section needs somewhat less computational burden, depending on required precedence relationships, but requires some logical operation so as not to violate given precedence restrictions. The blocking of subsequences is a way of introducing the precedence restrictions into what is otherwise an assignment problem and is accomplished rather successfully by sequential multiplication in Algorithm 1 and by using the fact in Algorithm 2, respectively. From a point of view of logical operation for precedence relationships, Algorithm 1 is more applicable and less burdensome, compared with Algorithm 2. Under loose precedence restrictions Algorithm 2 may be more useful than Algorithm 1.

2.4 SEQUENCING A SET OF JOBS TO MINIMIZE THE TOTAL DEFERRAL COST ASSOCIATED WITH COMPLETION TIMES WITH PRECEDENCE RESTRICTIONS

2.4.1 Problem Statement

The problem is characterized as follows : There are n jobs, precedence relationships imposed on them, and a facility to process them. For each job i $(i=1,2,\ldots,n)$, there is a fixed processing time p_i that does not depend upon which jobs precede or follow the job on the facility. Associated with job i is a deferal cost $c_i(t_i)$, where t_i is the flow time of the job. It is desired to find a feasible sequence for the jobs such that the total deferral cost,

$$\sum_{i=1}^{n} c_i\ (t_i)$$

is as small as possible.

The problem discussed in Section 2.2 is the special case of this :

$$c_i(t) = u_i t \quad , \quad u_i \geqq 0 \qquad (i = 1,2,\ldots, n) \; .$$

2.4.2 Algorithm

In what follows, it is assumed further that the processing-time of each job is 1. If precedence relationships are not imposed on the jobs, then the problem reduces to an assignment problem and therefore it can be solved quite easily. If the jobs have required precedence relationships and deferral costs are nonlinear but monotonically nondecreasing with time, and if max $c_i(t_i)$ is to be minimized, then the problem can also be solved in calculation of polynomial order of n. But as far as the auther knows, no effective method for the above problem except a dynamic programming method which requires about $n2^n$ arithmetic operations without regard to required precedence relationships has been proposed so far. In this section the problem is tackled by one of the previous methods. Since the processing-time of each job is 1, make an (n x n) square matrix and record $a_{p,q}$ in the column corresponding to the job q in the row corresponding to the completion time p of the job q . For the same precedence diagram shown in Fig. 2.11, consider the deferral costs shown in Fig. 2.21 .

Operations / Time	1	2	3	4	5	6	7	8	9	10
1	11									
2		6	5							
3		9	6	20	24	15	31			
4		11	7	24	61	19	32			
5		53		28	74	23	33			
6		88		33	80	29	34	8		
7				40	99	40	35	34		
8				50			36	68	62	
9								82	83	
10										42

Fig.2.21. Deferral costs associated with the completion times of the jobs .

Since precedence relationships are imposed in addition to the completion times of the jobs, Algorithm 2 is almost unapplicable. Logical operations for not violating precedence relationships are too complecated to be handled.

Sequential multiplication applies correspondingly to the process of branching when branching is required further after the first branching. The different consideration from Algorithm 1 should be paid to take completion times and required precedence restrictions into account simultaneously.

This is settled by introducing the concept of ' the degree of freedom ' that indicates the interval in which the completion time of a job appears. This point is the essential difference between Algorithm 1 and the algorithm developed in the following. The calculation of the degree of freedom of a job can be done by the following equation :

The degree of freedom of a job = The total completion time of all jobs - the number of preceding jobs of the job - the number of following jobs of the job.

For example, in the figure, jobs 1 and 10 have 1 degree of freedom since job 1 should be processed first and job 10 last. Job 2 has 5 degrees of freedom since the preceding job of job 2 is 1, and the following jobs are 4, 8, 9, and 10, and therefore $10 - 1 - 4 = 5$. The earliest completion time of job 2 is 2 and the latest completion time is 6 .

Since the process of branching follows sequential multiplication, checking the decrease of degrees of freedom after branching can be restricted to the decrease of the degree of forward freedom. Suppose that (i,j) is selected as the i th knot and that after crossing off i th row, j th column, jobs h_1, h_2, ... , and h_k are in the $(i+1)$ st row. If there are precedence relationships among h_1, h_2, ..., and h_k , then the following jobs among them can not be completed at time $(i + 1)$ and therefore corresponding values are set to ∞. By this simple operation checking the decrease of degrees of freedom is performed. Now the algorithm to solve the problem can be developed.

<u>Algorithm 3</u> :

Step 1-A. For a given precedence diagram, calculate the degrees of freedom for each job in a given set of jobs $X = \{1, 2, \ldots, n\}$, and then construct the deferral cost matrix L_0. Draw a knot 'Γ_1' in a tree-like diagram.

Step 1-B. Reduce L_0 and obtain L_1. Apply the assignment model optimization to L_1 and obtain L_1'. Get the first solution, and calculate the lower bound $l(\Gamma_1)$. Give the value to knot 'Γ_1'.

Step 1-C. Check by sequential product whether or not the solution satisfies required precedence relationships. If feasible, stop, since the solution is an optimal feasible sequence ; Otherwise let $\pi = 1$ (assumed that the first job to be processed is 1) and go to Step m-A.

Step m-A, $(m \geqslant 2)$. Extend the selected partial sequence π, from job i which is the last job in π, to job j which are reached from i by direct transition. Make branches of knot '(m,j)'.

Step m-B. Perform for each of the jobs j the following (i), (ii), and (iii) :

(i) Cross off row m, column j from the matrix $L_{m-1,\bar{i}'}$, where suffix (m-1) shows a matrix produced at Step (m-1), and in addition to this, i should be added for distinction since generally more than one matrix is produced at Step (m-1).

(ii) Check the decrease of the degrees of freedom for each job in the row (m+1). The jobs which can not be processed at time (m+1), set the values associated with them to ∞ .

(iii) Reduce the matrix and obtain the reduced matrix $L_{m,j}$. Apply the assignment model optimization to $L_{m,j}$ and obtain $L_{m,j}'$. Obtain the solution, and calculate the lower bound.

Give the value to knot '(m,j)'.

(iv) Select the solution associated with (m,j) which gives the minimum lower bound.

Step m-C. Check by sequential product whether or not the solution satisfies the required precedence relationships.

① If feasible, terminate branching thereafter from knot '(m,j)'

(i) If the lower bound of knot '(m,j)' does not exceed any lower bound of the unbranched knots, stop, since the solution is an optimal feasible sequence ;

(ii) Otherwise, select the unbranched knot which has the minimum lower bound.

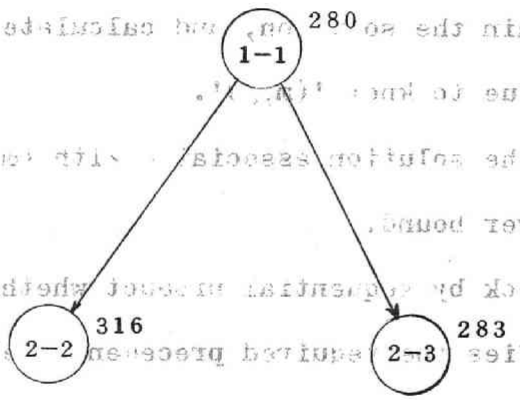
(A) If the knot is terminated, stop, the solutions is an optimum feasible sequence ;

(B) Otherwise, go to Step m-A.

② If not feasible, go to Step m-A.

If the step are carried to a sufficient degree, an optimum feasible sequence will eventually be produced.

For the example shown in Fig. 2.21, the application of the algorithm Step 1-A and B results in an infeasible solution and therefore branching is needed. The result of carrying out branching far enough is shown in Fig. 2.22.



$\pi = 1.3.5.2.6.8.4.7.9.10$

Fig.2.22. Tree-like diagram.

Now, assume for the sake of generality that the processing-time of each job is not necessarily one, but an arbitrary integer. Lawler[26] showed that the case in which no precedence relationships are imposed can, sometimes, be solved by the transportation algorithm. His method is not generally applicable at all. In what follows it is going to be shown that Algorithm 3 can be extended to this more general case easily.

Suppose the total processing-time for performing all the jobs is n, and make an (n x n) square matrix. Assign each job as many columns as the processing-time of the job, and as many rows as the difference between the earliest starting time and the latest completion time. The earliest starting time is calculated by summing up the processing-times of precedence jobs, the latest completion time, the processing-times of following jobs. For instance if the processing-time of a job is i, the earliest starting time is k, the latest completion time is l, then $\{l-(k-1)\}$ rows and i columns are assigned to the job. Deferral costs are written in the boxes in the last column of i column from $(k-1+i)$th row to l th row. For each box having a deferal cost, assign 0 to the boxes which stand on the diagonal from first to $(i-1)$ st columns.

The deferral cost matrix is prepared for the given jobs in this manner. In order to obtain lower bound on times, the following assumption is established : In applying the assignment model optimization, suppose that a certain box is chosen as the completion time of a job. Then the 0's on the diagonal to the box in the submatrix assigned to the job must always be chosen. Under this assumption the assignment model optimization might not be able to be completed. This does not cause much inconvenience. In such a case having a slightly worse lower bound should satisfy. From the above discussion the algorithm to solve the general problem can be developed. But it is almost the same as Algorithm 3 and therefore is omitted to avoid repetition.

2.4.3. Discussions

The problem of sequencing a set of jobs to minimize the total deferral cost associated with completion times with precedence restrictions can be solved by a dydamic programming approaching. The similar formulation as the formulation for solving the traveling salesman problem can be made, subject to precedence restrictions which are handled by considering sequential product. The method developed in this section can handle more jobs than the dynamic programming approach can. It also makes use of the information which unassigned jobs have, while the dynamic programming approach does not, but uses only the information which assigned jobs have.

2. 5 CONCLUSIONS

This chapter dealt with three sequencing problems whose objective functions are strongly related to minimizing the costs of inventory, facility utilization and lateness, respectively.

For the problem of sequencing a set of jobs to minimize mean weighted flow-time with precedence restrictions, associated with minimizing the cost of inventory, several effective theorems were developed to reduce the existential range of optimum solutions. It is surmised that these theorems should be used as far as they can apply and that if they are not applicable, given precedence relationships should be rewritten by adding more precedence constraints, whose technique is somewhat similar to the branch and bound technique.

For the problem of sequencing a set of jobs to minimize the sum of sequence-dependent setup-times with precedence restrictions, associated with minimizing the cost of facility utilization, two algorithms were developed, to one of which the consideration of not violating given precedence relationships was given more weight than that of obtaining a good lower bound and to another, vs. Anyhow two algorithms are based on the method of establishing linear sequences, and the branch and bound technique. The first algorithm is more general subject to precedence relationships.

For the problem of sequencing a set of jobs to minimize the total deferral cost associated with completion time with precedence restrictions, with the aim of minimizing the cost of lateness, it has been shown that the first algorithm for the second problem can also be applied to this problem with a slight modification.

CHAPTER 3 ESTABLISHMENT OF COMPOUND SEQUENCES

3. 1 Problem Definition

The purpose of this chapter is to find a systematic method to establish all of the feasible compound sequences which are composed of feasible subsets of operations. In compound sequences a feasible subset of operations is assumed to be performed at a time. Establishment of compound sequences of linear type can be done by using linear product which was introduced in Chapter 1. In this chapter the following method will be established for several reasons.

In order to establish feasible complete compound sequences, feasible subsets of operations will be produced as elements of compound sequences first, and then the systematic way of establishing all of the feasible compound sequences in terms of feasible subsets of operations will be discussed. The main reason that the method will be introduced is that the establishment of compound sequences in terms of feasible subsets of operations is more effective rather than that of compound sequences in terms of operations. Also various practical restrictions such as positional and combinatorial restrictions are easily imposed on subsets of operations, otherwise they are burdensome and difficult to be handled.

Mainly the following are considered :

(1) A feasible subset of operations and its precedence relationships.

(2) The suitable way of constructing feasible subsets of operations without overlapping.

(3) The systematic way of establishing compound sequences. After several definitions and notations are introduced, the first problem is tackled in Section 3.3 to develop a subsequential product. Based on the result of Section 3.3 the second problem is considered in Section 3.4 to introduce a combinatorial matrix approach. Moreover the consideration of minimum-time arrays for compound sequences of overlap type results in the formulation of several effective theorems for finding minimum-time arrays in the section. Then in Section 3.5 the third problem considered is that of development of subsequential multiplication. This leads to easy and systematic establishment of compound sequences.

3.2 DEFINITIONS AND NOTATIONS

Compound sequences may be classified into the following two types as illustrated in Fig. 3.1, according to the procedure of processing a subset of operations.

(1) Compound sequences of linear type

An individual operation in a subset of operations is processed one at a time.

(2) Compound sequences of overlap type

An individual operation in a subset of operations can be processed simultaneously with other operations in the subset.

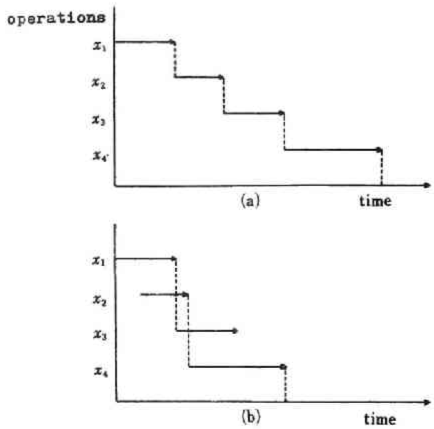


Fig.3.1. (1) Compound sequence of linear type,
(2) Compound sequence of overlap type.

Of course, type (1) can be said to be a special case of type (2). Establishment of type (2) is much more difficult than that of type (1). This chapter develops the common method for establishing type (1) and type (2).

The following definitions and notations are used in addition to those introduced in the previous chapters :

$x_i \lesseqgtr x_j$: There exists a precedence relationship between x_i and x_j .

$x_i \not\lesseqgtr x_j$: There exists no precedence relationship between x_i and x_j .

Transition of type 1 : If there exists an arrow from x_i to x_j , the transition is called the transition of type 1 and is denoted by $c_t(x_i, x_j) = I$.

Transition of type 2 : If there exists no arrow from x_i to x_j , the transition is called the transition of type 2 and denoted by $c_t(x_i, x_j) = II$.

. $P\{x_i\} := \{x_j \mid x_i \gg x_j, x_i, x_j \in X\}$.

A set of operations which directly precede x_i .

. $F\{x_i\} := \{x_j \mid x_i \ll x_j, x_i, x_j \in X\}$.

A set of operations which directly follow x_i .

. $P_A\{x_i\} := \{x_j \mid x_i \geqslant x_j, x_i, x_j \in X\}$.

A set of all operations which precede x_i .

. $F_A\{x_i\} := \{x_j \mid x_i \leqslant x_j, x_i, x_j \in X\}$.

A set of all operations which follow x_i .

. λ : A subset of operations. Subsets of operations of linear type make a one-dimensional array. Subsets of operations of overlap type, a multi-dimensional array. "A_r" is used to denote an explicit array if necessary. In case of multi-dimensions, a subset of operations arrayed in a row (column) is called a row (column) array and is denoted by α_i (β_j). In general the cardinal numbers of row arrays in an array are different, the subset of the operations which stand in the first (last) place in the rows is called initial (last) column array.

. $\mu(x_i)$ under Ar :

A subset of operations which can be processed next from x_i which belongs to the last column array of Ar.

. ${}^m\lambda_i$:

The letter m represents the cardinal number of the subset λ, viz., the number of operations which construct λ_i .

In case of necessity for distinction, i may be added.

In some cases, i represents the i th position of a compound sequence.

. ξ : A compound sequence.

. ${}^m\xi_i$:

The letter m represents the cardinal number of the sequence ξ_i , viz., the number of subsets of operations which construct ξ_i . In case of necessity for distinction, i may be added.

. ${}^m\xi_i(\lambda_1, \lambda_m)$:

The compound sequence of cardinal number m which processes from λ_1 to λ_m .

. $S\{\theta\}$:

A set of operations which are contained in θ . The letter θ may denote either an operation, or a subset of operations, or a compound sequence.

. $^{\forall}x(x \in S\{\theta\})$:

For x, the proposition : $x \in S\{\theta\}$ is valid.

. $t(\theta)$: The time necessary to process θ .

. $t(x_i \ldots x_j)$ in Ar : The time necessary to perform the row array $(x_i \ldots x_j)$ in Ar.

3.3 CONSTRUCTION OF FEASIBLE SUBSETS OF OPERATIONS

First, construction of subsets of operations of linear type is discussed.

3.3.1 Linear Type

(1) A subset of operations and its precedence relationships. A subset of operations is, in general, composed of operations, but an operation is also assumed to be a subset of operations for the sake of convenience. If the total number of operations is n, the total number of subset of operations is theoretically $\sum_{i=1}^{n} {}_nC_i = 2^n - 1$. The feasible subsets of operations may decrease due to required precedence relationships, various technological restrictions and so forth. In what follows, it is assumed that a transition from a subset of operations to another is possible only when it is not breaking the precedence relationship between two subsets and only after all of the operations of the preceding subset of operations have been performed. Under this assumption, the following subset of operations becomes infeasible, viz.,

the subset of operations on some of which precedence relationships are imposed through an operation that is not contained in the subset of operations. It must be crossed off from a set of feasible subsets of operations. The subsets of operations which do not satisfy various other technological restrictions must be crossed off from the set of feasible subsets of operations.

In what follows, a subset of operations and its precedence relationships will be discussed, assuming that it is feasible, and then how to treat other various technological restrictions will be considered.

In order to perform a subset of operations without violating given precedence relationships, all of the opera tions that precede each operation contained in the subset must be performed in advance. Hereupon the following definitions are introduced for a subset of operation λ :

- $P\{\lambda\} = \{x_j \mid x_j \geqslant x_k,\ x_j,\ x_k \in \bigcup_{x_i \in \lambda} P\{x_i\},\ x_j \notin \lambda\}$ (3.1)
- $F\{\lambda\} = \{x_j \mid x_j \leqslant x_k,\ x_j,\ x_k \in \bigcup_{x_i \in \lambda} F\{x_i\},\ x_j \notin \lambda\}$ (3.2)
- $P_A\{\lambda\} = \{x_j \mid x_j \in \bigcup_{x_i \in \lambda} P_A\{x_i\},\ x_j \notin \lambda\}$ (3.3)
- $F_A\{\lambda\} = \{x_j \mid x_j \in \bigcup_{x_i \in \lambda} F_A\{x_i\},\ x_j \notin \lambda\}$ (3.4)

$P\{\lambda\}$ represents a set of operations which directly precede the operations contained in λ and which do not belong to λ ;

$F\{\lambda\}$, a set of operations which directly follow the operations contained in λ and which do not belong to λ ; $P_A\{\lambda\}$, a set of all operations which precede the operations contained in λ and which do not belong to λ ; $F_A\{\lambda\}$, a set of all operations which follow the operations contained in λ and which do not belong to λ , respectively.

The precedence relationship of λ can be represented by $P\{\lambda\}$, $F\{\lambda\}$, $P_A\{\lambda\}$ and $F_A\{\lambda\}$. $P\{\lambda\}$ reduces to $c(i,j)$ in case of linear sequence and is usually good enough to show the precedence relationships of λ.

In the above discussion, the feasibility of subset λ was assumed. Now, it is time to clarify how to construct feasible subsets of operations.

(2) Subsequential Product .

It will be considered on what conditions an operation x_j which is not contained in the subset can be added to a subset of operations ${}^m\lambda_i$.

For an arbitrary operation x_k contained in ${}^m\lambda_i$ and x_j :

(I) If there exists no precedence relationship between x_k and x_j , x_j can be added to ${}^m\lambda_i$, and make a new subset of operations ${}^{m+1}\lambda_{i'}$. This is denoted by ${}^m\lambda_i \cdot x_j = {}^{m+1}\lambda_{i'}$.

(II) If there exists a precedence relationship between x_k and x_j , and if x_j belongs to either $P\{{}^m\lambda_i\}$ or $F\{{}^m\lambda_i\}$, that is, if x_j does not precede or follow x_k indirectly through other operation or operations which are not contained in ${}^{m}\lambda_i$, x_j can be added to ${}^{m}\lambda_i$, and makes a new subset of operations ${}^{m+1}\lambda_{i'}$.

(III) Otherwise x_j cannot be added to ${}^{m}\lambda_i$.

This insists that x_j has a direct transitional relationship with an operation in ${}^{m}\lambda_i$ if x_j can be added to it. The conditions on direct transition are represented by a fundamental matrix which was introduced in Chapter 1. Direct transitional relationships between an operation x_j , and an operation x_k in ${}^{m}\lambda_i$ will be represented by the fundamental matrix. For this purpose it is desired to fix the order of the operations in ${}^{m}\lambda_i$. It is also necessary for constructing feasible subsets of operations without overlapping. In what follows, it is assumed that the order of the operations in ${}^{m}\lambda_i$ is subject to the order of their ranked numbers. Under this assumption, the operations directly transitionable from the last element in the permutation of a subset of operations are picked up from the fundamental matrix. Suppose that the last element in ${}^{m}\lambda_i$ is x_i . Row x_i in the fundamental matrix shows the operations which directly precede x_i , $F\{x_i\}$, and the operations which have no precedence relationships with x_i . Of all the operation directly transitionable from x_i , what operation can be added to make a new subset of operations ? After careful consideration, the conditions under which an operation can be added to ${}^{m}\lambda_i$ are summarized as follows :

(I) If x_j is included in $S\{{}^{m}\lambda_i\}$, x_j can not be added to ${}^{m}\lambda_i$.

(II) If $c(x_i, x_j) = \phi$, x_j can be added to ${}^{m}\lambda_i$, and makes a new subset of operations ${}^{m+1}\lambda_{i'}$. In this case

$$P\{{}^{m+1}\lambda_{i'}\} = P\{{}^{m}\lambda_i\} \tag{3.5}$$

(III) Let

$$c(x_i, x_j) - S\{{}^{m}\lambda_i\} = c'(x_i, x_j) \tag{3.6}$$

then if $c'(x_i, x_j) = \phi$, x_j can be added to ${}^{m}\lambda_i$, and makes a new subset of operations ${}^{m+1}\lambda_{i'}$ since the operations which must be processed earlier than x_j to process x_j are included in $S\{{}^{m}\lambda_i\}$. In this case (3.5) also holds.

(IV) If $c'(x_i, x_j) \neq \phi$, and if

$$x(x \in c'(x_i, x_j)) \leqslant \text{each of the operations included in } S\{{}^{m}\lambda_i\} \tag{3.7}$$

then x_j can be added to ${}^{m}\lambda_i$ and makes a new subset of operations. Relation (3.7) means that x_j belongs to $F\{{}^{m}\lambda_i\}$. In this case,

$$P\{{}^{m+1}\lambda_{i'}\} = \{x_k | x_k > x_l;\ x_k, x_l \in P\{{}^{m}\lambda_i\} \cup c'(x_i, x_j)\} \tag{3.8}$$

(V) Otherwise, x_j can not be added to ${}^{m}\lambda_i$.

Taking these into consideration and arranging the process of constructing a feasible subset of operations from lower to higher cardinal number with the fundamental matrix, results in the definition of subsequential product whose flow chart is shown in Fig. 3.2.

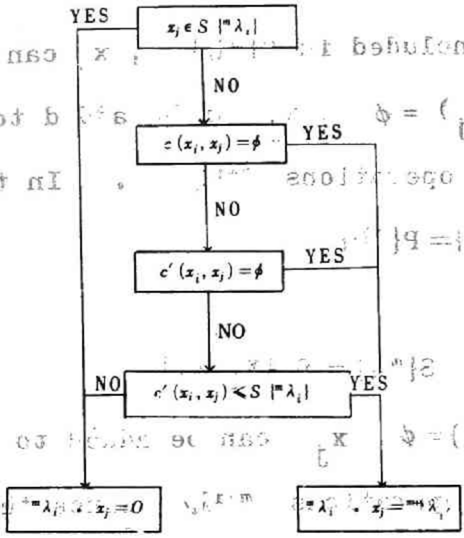


Fig.3.2. Subsequential product .

Now it is time to consider how to construct feasible subsets of operations systematically without overlapping by subsequential product.

(3) Combinatorial matrix

To consider the subsequential product between $^m\lambda_i$ and x_j systematically, a combinatorial matrix is introduced, in which operation x_j is arranged in column in the ranking order and a subset of operations $^m\lambda_i$ in row.

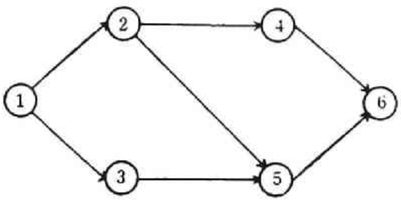


Fig.3.3. Precedence diagram.

		1	2	3	4	5	6
	1		2	3			
1	2			3	4	3) 5	
1	3				2) 4	2) 5	
2	4					3) 5	5) 6
2,3	5						4) 6
4,5	6						
	1.2			1.2.3	1.2.4		
	1.3						
1	2.3				1) 2.3.4	1) 2.3.5	
1	2.4					3) 2.4.5	
3	2.5						
2	3.4					2) 3.4.5	
2	3.5						4) 3.5.6
2,3	4.5						2,3) 4.5.6
5	4.6						
3,4	5.6						
	1.2.3				1.2.3.4	1.2.3.5	
	1.2.4						
1	2.3.4					1) 2.3.4.5	
1	2.3.5						
3	2.4.5						3) 2.4.5.6
2	3.4.5						2) 3.4.5.6
4	3.5.6						
2,3	4.5.6						

Fig.3.4. Illustration of a combinatorial matrix.

Attention should be paid to prevent the same subsets of operations from being produced. For the purpose, only the subsequential product between ${}^{m}\lambda_i$ and the operation x_j whose ranked number is higher than the number of any of the operations which construct ${}^{m}\lambda_i$ should be considered. For the example whose precedence diagram is shown in Fig. 3.3, the combinatorial matrix lists operations 1,2,3,4,5, and 6 in row and in column according to their ranked orders as illustrated in Fig. 3.4. Then subsets of operations of cardinal number 2, i.e., ${}^{2}\lambda_i$, are constructed in the upper triangular matrix. Resultant subsets of operations are successively added in the extended rows. Then subsets of operations of cardinal number 3 are constructed and so forth. In the figure, subsets of operations up to cardinal number 3 are constructed. The first column shows the operations which must be processed earlier than the subsets of operations listed in the second column, called the precedence operations. The operations shown in the upperwestern corner in each row and in each column show the precedence operations associated with the subset of operations in the box. For example, look at the third row. The results of operations constructed from this row are 2.3, 2.4, and 2.5. Their precedence operations are 1,1, and 3 by equation (3.5) and (3.8). Now, look at the eighth row. The subset of operations listed in the row and in the second column is 1.2. Since the transitions from 2 to 3, and 4 are possible unconditionally, the subsets of operations, whose cardinal number is 3, 1.2.3 and 1.2.4 are constructed, respectively. Although the transition from 2 to 5 is possible subject that operation 3 is processed earlier than 5, subset of operations 1.2.5 can not be constructed since the transition does not satisfy the condition. Subsets of operations of high cardinal number are successively constructed in the same manner.

(4) A Subset of Operations and Other Restrictions .

From economical or technological point of view, various other relationships are imposed on subsets of operations. They also reduce the number of possible subsets of operations. For instance, in assembly lines the processing-time given to a station that performs a subset of operations is limited. These restrictions are also taken into consideration at the same time when the subsequential product of operations in a combinatorial matrix is considered.

3.3.2 Overlap Type

(1) The arraying problem .

In case of linear type, the array of operations in a subset of operations can be fixed according to their ranked numbers subject that the transition-time is sequence-independent, while in case of overlap type, the array of operations in a subset of operations comes into question since the processing-time of a subset of operations depends on the way of arraying the operations in the subset. In what follows, subsequences which involve either preemption or inserted idle-time are not considered. In what way multi-dimensional arrays should be obtained and an array which gives a minimum time, called a minimum time array, can be found ? Two procedures can be considered to construct a subset of operations of overlap type :

① The first procedure pays much attention to the consideration of precedence relationships. As an initial condition, the first column array is given and then operations are added to it successively without violating given precedence relationships. The minimum-time array is determined among the arrays which include the same operations. As its modification, if all of the operations which construct a subset of operations are first given, then the minimum-time array can be determined more easily.

② It is well speculated that the number of feasible subsets of operations may be enormous as the number of operations increases. In order to reduce the total number of possible combinations, row arrays are first constructed as elements of multi-dimensional arrays beforehand and then they are combined without disturbing precedence relationships to construct multi-dimensional arrays. The minimum-time array is considered at the time of combining row arrays.

In ① the consideration of precedence relationships is given more weight, and in ② the consideration of reducing possible combinations. After careful consideration, it is concluded that procedure ② should be rejected. The demerit of procedure ② is that the logical operations to combine row arrays are too complicated to be handled due to required precedence relationships, and therefore require enormous memory. Since by this procedure arrays which include the same operations and which have different row arrays are produced, finding minimum-time arrays is also troublesome. For this discussion, multi-dimensional arrays are constructed by procedure ① in this dissertation. For the sake of explanation the modification of procedure ① is adopted in the following. The difference of procedure ① and its modification is that the latter requires easier comparison to get the minimum-time array. The operations of obtaining multi-dimensional arrays are the same in both the procedures. To obtain multi-dimensional arrays by subsequential product using a fundamental matrix requires the following consideration, i.e., it is necessary to know the conditions of indirect transition. For the example shown in Fig. 3.5, the fundamental matrix with the conditions of indirect transition is given in Fig.3.6.

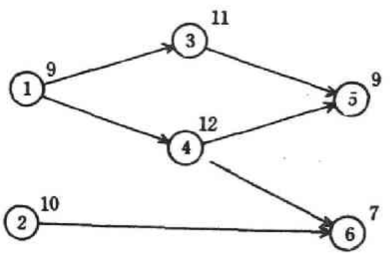


Fig.3.5. Precedence diagram with processing-times.

	1	2	3	4	5	6
1		2	3	4	$^{3,4)}\underline{5}$	$^{2,4)}\underline{6}$
2	1		$^{1)}3$	$^{1)}4$	$^{3,4)}5$	$^{4)}6$
3		2		4	$^{4)}5$	$^{2,4)}6$
4		2	3		$^{3)}5$	$^{2)}6$
5		2				$^{2)}6$
6			3		$^{3)}5$	

Fig.3.6. Fundamental matrix with the conditions of indirect transition .

In the figure, the conditions of indirect transition are shown by drawing underlines. In case of multi-dimensional arrays indirect transitions are possible, and therefore the operations which can be added to an operation in the last column array can also be added to any operation is the last column array.

In case of p-dimensional array, given p-dimensional initial column array all of the feasible subsets of operations can be obtained systematically using subsequential product in the following manner. The operations which can be added to the i th row of the p-dimensional initial column array are the ones which are transitionable from the operation of the i th row of the p-dimensional initial column array and which require as precedence operations no operations except the p operations of the initial column array. Subsets of operations of high cardinal number can be obtained successively in this manner. But this procedure leads to a trouble of producing subsets of operations with overlapping. Avoiding this is the essential matter to construct feasible subsets of operations.

<u>The way to avoid overlapping.</u>

Suppose that the rows in a combinatorial matrix are extended up to subsets of operations of cardinal number m . In the array $\mathrm{Ar}\,1 = \{ \cdots / \mathrm{p} \cdots \mathrm{x} / \cdots \}$, suppose z can be added to x. (each portion divided by / denotes a row array). Overlapping occurs if the row array $\overline{\mathrm{p} \cdots \mathrm{x} \cdot \mathrm{z}}$ has been already constructed before in the steps listed before A_r 1. Therefore in this case the array

$Ar\ 2 = \{ \cdots / \ p \ldots x.z \ / \cdots \}$ should be rejected. By this consideration, overlapping of producing the same subsets of operations can be avoided and therefore all of the feasible subsets of operations are constructed.

(2) Minimum - time arrays.

In case of finding minimum-time arrays, the difference of procedure ① and its modification is that the latter requires easier comparison to get minimum-time arrays, but at the same time it requires that the operations which construct a subset of operations should be fixed beforehand. To find minimum-time arrays there is no essential difference between these, and the same theorems to reject dominated arrays are used. In what follows, the modification of procedure ① is assumed for the sake of explanation.

What transitions are superior to others and what arrays dominate others should be considered in order to find effective theorems to discard inferior arrays. To answer the former results in the formulation of the following theorem.

Dominance Theorem 3.1 In a fundamental matrix, suppose that transitions from x to y, and from y to x are possible. If $P_A(x) \in P_A(y)$ and $F_A(x) \ni F_A(y)$, then the time of the row array which includes x.y does not exceed that of row array which has y.x . Therefore in this case x.y dominates y·x .

(Proof)

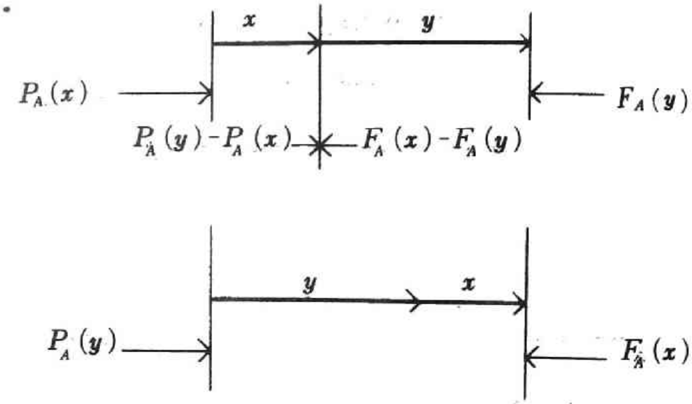


Fig.3.7. Proof of Theorem 3.1 .

This is the problem of possibility for row arrays x·y and y·x to have shorter time in performing. First consider the condition $P_A(x) \in P_A(y)$. As shown in the upper part of Fig. 3.7, in order to process x.y , $[P_A(y)-P_A(x)]$ should be performed by the completion time of x, while in the lower part of Fig. 3.7, $P_A(y)$ should be processed by the completion-time of y. Therefore the array which includes x·y has the possibility of becoming more compact than the one which includes y·x . The same argument can be applied to $F_A(x) \ni F_A(y)$. ||

The application of the theorem to a fundamental matrix makes it easier to be handled. For example, Fig. 2.8 resulted from Fig. 2.6. In the figure, the elements crossed off by X are the ones deleted by the theorem. By the application of the theorem it can not be said that the operation which can be added to an operation in the last column array can also be added to any operations in the last column array.

Identification of dominant arrays will be discussed next.

	1	2	3	4	5	6
1		2	3	4	3,4) 5	2,4) 6
2	X		1) 3	1) 4	3,4) 5	4) 6
3		2		X	4) 5	2,4) 6
4		2	3		3) 5	2) 6
5		X				2) 6
6			X		3) 5	

Fig.3.8. Application of Theorem 3.1.

Dominance Theorem 3.2. In the array Ar3 = { .../p...x/... /q...y/... }, suppose that z can be added to x and y and that $t(p...x) > t(q...y)$. If $x \in c(y,z)$, then the array Ar 4 = {../p...x.z/.../q...y/..} dominates the other.

(Proof) As illustrated in Fig. 3.9 (b) the broken line portion becomes idle, while in (a) there is the possibility that the portion can be used. Arrays (a) and (b) affect thereafter the same way except the broken line portion. Therefore array (a) dominates the other.

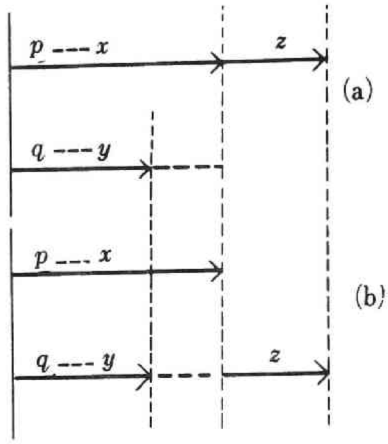


Fig.3.9. Proof of Theorem 3.2.

Corollary. In the array $A_r 5 = \{.../p...x/.../q...y/...\}$, suppose that u and v can be added to x and y and that $t(p...x) > t(q...y)$. If either $c(x,u) = \phi$ or $c(x,u)$ y, and if $c(y,v) \ni x$, then $Ar6 = \{../p...x.v/.../q...y.u/..\}$ dominates the other.

(Proof) The application of the same argument as the proof of Theorem 3.2 to illustrated Fig. 3.10 results in the corollary.

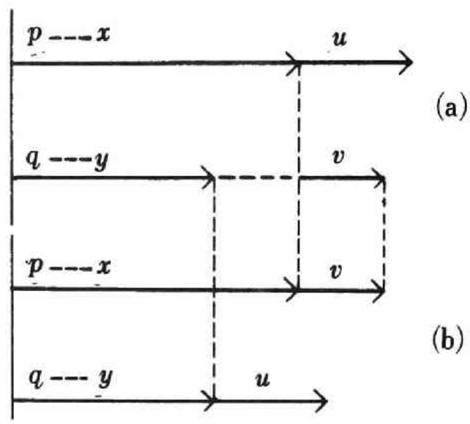


Fig.3.10. Proof of the corollary.

Dominance Theorem 3.3. Under the same assumptions as Theorem 3.2, if $\mu(x)$ under Ar 3 $= \mu(y)$ under Ar 3 $=$ z then Ar 7$=\{\ldots/p\ldots x/\ldots q\ldots y.z/\ldots\}$ dominates the other.

(Proof) As illustrated in Fig. 3.11, only Z can be added to y from the assumptions. Even if some operation can be added to p...x.z as in (a), the portion of the bold broken line becomes idle. In (b) or (c) idle portions do not exceed the idle portion of (c), and therefore Ar 7 dominates the other.

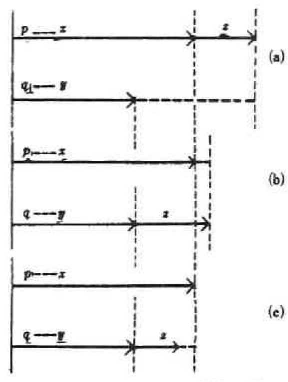


Fig.3.11. Proof of Theorem 3.3. and 3.4.

Dominance Theorem 3.4. Under the same assumptions as Theorem 3.2, if $\mu(y)$ under Ar 3 = z and $c_t(y,z) = \text{I}$, then Ar 8 $=\{\ldots/p\ldots x/\ldots/q\ldots y.z/\ldots\}$ dominates the other.

(Proof) As illustrated in Fig. 3.11, array (a) has the idle portion shown by a bold broken line since the transition from y to z is type 1. Arrays (b) or (c) has the less idle portion than (c). Therefore Ar 8 dominates the other.

To use Theorem 3.4 effectively the fundamental matrix should indicate which type transitions are. In Fig. 3.6, underlines are drawn to the cases of type I .

Dominance Theorem 3.5. In the array Ar 9 $=\{\ldots/p\ldots x\ldots z/\ldots/q\ldots y/\ldots\}$ suppose w can be added to z and y and that

$$t(p\ldots x) > t(q\ldots y).$$

If $\mu(y)$ under Ar 9 $=\{a,b,\ldots c\}$ and if $c(y,a)\ni x, c(y,b)\ni x, \ldots, c(y,c)\ni x$ then the array Ar 10 $=\{\ldots/p\ldots x\ldots z.w/\ldots/q\ldots y/\ldots\}$ is dominated.

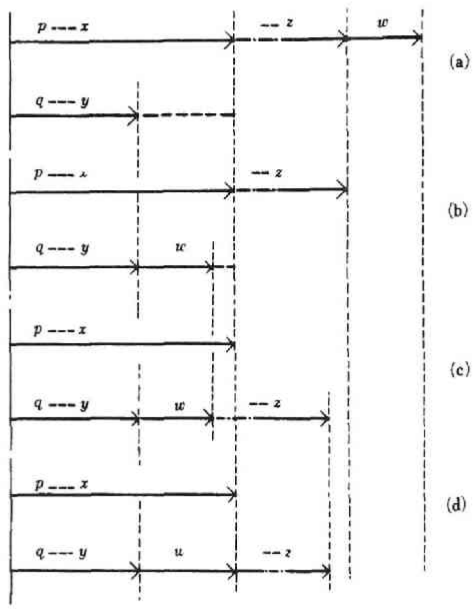


Fig.3.12. Proof of Theorem 2.5.

(Proof) As illustrated in Fig. 3.12. (a) the bold broken line portion becomes idle. In the case (b), where $t(p\ldots x)>t(q\ldots y)+t(w)$, the idle part of (b) is less than that of (a). In the case (c), where (... z) can be added to q ... y . w without idle time, the idle portion of (c) is still less than that of (b). In the case (d), where $t(p\ldots x) \leq t(q\ldots y)+t(w)$, if $(\ldots z)$ is added to q ... y . w as in (c), less idle time is obtained. Therefore (a) is dominated.

Dominance Theorem 3.6. Suppose that there are two arrays Ar 11, and Ar 12 whose last column arrays are β_1, and β_2, respectively, and that

$$S\{Ar\,11\} = S\{Ar\,12\} \quad \text{and} \quad S\{\beta_1\} = S\{\beta_2\}.$$

For the completion times of x_i belonging to β_1, and those of x_i belonging to β_2, if

$$t(\ldots x_i) \text{ in Ar 11} < t(\ldots x_i) \text{ in Ar 12},$$

then Ar 11 dominates Ar 12.

(Proof) Self-evident .

By the theorems the existential range of the minimum-time array can be reduced. Then, the minimum-time array should be found among resultant arrays.

In the final analysis it will be considered how to give initial column arrays. In a one-dimensional array, there is no problem since each operation can be an initial operation. In an p-dimensional array, p parallel operations should be given as an initial column array, but due to given precedence relationships, sometimes only less than p parallel operations can be found. In this case, the portion which has no operations as an initial operation becomes idle. On the other hand, if there are conceivably more than p operations as initial operations in a subset of operations, there are many ways of choosing p initial operations. For this case the following theorem can reduce unnecessary inferior combinations.

Dominance Theorem 3.7. Suppose that in a p-dimensional initial column arrays β_3 and β_4, (p-1) operations are common to β_3 and β_4 and that β_3 has x which is not included in β_4, and that β_4 has y which is not included in β_3. If $t(x) \geqslant t(y)$ and if $F_A(x) \ni F_A(y)$, then β_3 dominates β_4.

(Proof) This is to be proved later in Chapter 4.

For the example shown in Fig. 3.7, the minimum-time arrays are found in the following way. Suppose that subsets of operations of two-dimensions are constructed. The initial column array is $\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}$. By the subsequential product whose chart is shown in Fig. 3.2, four subsets of operations whose cardinal number is 3 are constructed as shown in Fig. 3.13.

	1	2	3	4	5	6
$\{^{1}_{2}\}$			$\{^{1.3}_{2}\}$ $\{^{1}_{2.3}\}$	$\{^{1.4}_{2}\}$ $\{^{1}_{2.4}\}$		
$\{^{1.3}_{2}\}$ $\{^{1}_{2.3}\}$ $\{^{1.4}_{2}\}$ $\{^{1}_{2.4}\}$			$\{^{1.4.3}_{2}\}$ $\{^{1.4}_{2.3}\}$ $\{^{1.3}_{2.4}\}$ $\{^{1}_{2.4.3}\}$	$\{^{1.3}_{2.4}\}$ $\{^{1.4}_{2.3}\}$		$\{^{1.4.6}_{2}\}$ $\{^{1.4}_{2}\|_{6}\}$ $\{^{1}_{2.4}\|^{6}\}$ $\{^{1}_{2.4.6}\}$
$\{^{1.3}_{2.4}\}$ $\{^{1.4}_{2.3}\}$					$\{^{1.3.5}_{2.4}\}$ $\{^{1.3}_{2.4.5}\}$ $\{^{1.4.5}_{2.3}\}$ $\{^{1.4}_{2.3.5}\}$	$\{^{1.3.6}_{2.4}\}$ $\{^{1.3}_{2.4.6}\}$ $\{^{1.4.6}_{2.3}\}$ $\{^{1.4}_{2.3.6}\}$
$\{^{1.3}_{2.4.5}\}$ $\{^{1.4}_{2.3.5}\}$ $\{^{1.3}_{2.4.6}\}$ $\{^{1.4}_{2.3.6}\}$					$\{^{1.3.5}_{2.4.6}\}$ $\{^{1.3}_{2.4.6.5}\}$ $\{^{1.4.5}_{2.3.6}\}$ $\{^{1.4}_{2.3.6.5}\}$	$\{^{1.3.6}_{2.4.5}\}$ $\{^{1.3}_{2.4.5.6}\}$ $\{^{1.4.6}_{2.3.5}\}$ $\{^{1.4}_{2.3.5.6}\}$

Fig.3.13. Illustration of obtaining the minimum-time array.

Since $\mu(1)$ under $\{^{1}_{2}\} = \{3,4\}$, subsets of operations $\{^{1.2}_{3}\}$ and $\{^{1}_{2.4}\}$ are constructed from $\{^{1}_{2}\}$. Since $\mu(2)$ under $\{^{1}_{2}\} =$ 3,4 , $\{^{1}_{2.3}\}$ and $\{^{1}_{2.4}\}$ are constructed. These four subsets are added to the extended rows. From $\{^{1.3}_{2}\}$ subset $\{^{1.3}_{2.4}\}$ is constructed since $\mu(3)$ under $\{^{1.3}_{2}\} = \phi$ and $\mu(2)$ under $\{^{1.3}_{2}\} = \{4\}$. From $\{^{1}_{2.3}\}$, $\{^{1.4}_{2.3}\}$ is constructed in the same manner. From $\{^{1.4}_{2}\}$ in the third column, $\{^{1.4.3}_{2}\}$ and $\{^{1.4}_{2.3}\}$ are constructed. The former is inferior to $\{^{1.4}_{2.3}\}$ by Dominance Theorem 3.5 since $\mu(2)$ under $\{^{1.4.3}_{2}\} = \{5,6\}$,

$c(2,5) = \{3,4\}$ and $c(2,6) = \{4\}$.

The latter is also crossed off from consideration of avoiding overlapping. From $\{^{1}_{4.2}\}$ in the sixth column $\{^{1.4.6}_{2}\}$ and $\{^{1.4}_{2}|_{6}\}$ are constructed, where $|$ means operation 6 can be processed after 4 has been done.

The latter is crossed off by Dominance Theorem 3.2. The former is also deleted from consideration of avoiding overlapping since $\mu(6)$ under $\{^{1.4.6}_{2}\} = \phi$, $\mu(2)$ under $\{^{1.4.6}_{2}\} = \{3\}$ and row array $\{2.3\}$ has been already constructed. Now look at the row of $\{^{1}_{2.4}\}$. Subset $\{^{1.3}_{2.4}\}$ dominates $\{^{1}_{2.4.3}\}$ since $\mu(1)$ under $\{^{1}_{2.4.3}\} = \{3,6\}$ and for $3 \in \mu(1)$ by Theorem 3.4, and for $6 \in \mu(1)$ [note $c(1,6) \rightarrow 2,4$] by Theorem 3.2. Subset $\{^{1.3}_{2.4}\}$ is crossed off from consideration of avoiding overlapping. In the case of arrays $\{^{1}_{2.4}|^{6}\}$ and $\{^{1}_{2.4.6}\}$, the latter dominates the former. The latter is also crossed off from consideration of avoiding overlapping since $\mu(1)$ under $\{^{1}_{2.4.6}\} = 3$, $\mu(6)$ under $\{^{1}_{2.4.6}\} = \phi$ and the row array 1.3 has been already constructed. Eventually subsets of operations $\{^{1.3}_{2.4}\}$ and $\{^{1.4}_{2.3}\}$ are left as subsets of operations of cardinal number 4. From these subsets of operations, subsets of operations of cardinal number 5 are constructed as shown in the figure. Among these, $\{^{1.3}_{2.4.5.6}\}$, $\{^{1.4}_{2.3.5.6}\}$, $\{^{1.3}_{2.4.5.6}\}$, and $\{^{1.4}_{2.3.6.5}\}$ are crossed off by Theorem 3.3 since they are dominated by $\{^{1.3.6}_{2.4.5}\}$, $\{^{1.4.6}_{2.3.5}\}$, $\{^{1.3.5}_{2.4.6}\}$, and $\{^{1.4.5}_{2.3.6}\}$ respectively. Subsets $\{^{1.3.6}_{2.4.5}\}$, and $\{^{1.3.5}_{2.4.6}\}$ are also deleted by Theorem 3.6 since they are dominated by $\{^{1.4.6}_{2.3.5}\}$, and $\{^{1.4.5}_{2.3.6}\}$, respectively.

In the final analysis, either $\left\{\begin{smallmatrix}1\cdot4\cdot6\\2\cdot3\cdot5\end{smallmatrix}\right\}$ or $\left\{\begin{smallmatrix}1\cdot4\cdot5\\2\cdot3\cdot6\end{smallmatrix}\right\}$ will be the minimum-time array. After easy calculation, it is concluded that both of these are the minimum-time array.

3. 4 ESTABLISHMENT OF COMPOUND SEQUENCES

The problem is to find collections of subsets of operations which satisfy the following three conditions :

(1) $\bigcup_i \lambda_i = X$

(2) $\lambda_i \cap \lambda_j = \phi \qquad (i \neq j)$

(3) If $x_i \in S\{\lambda_i\}$, $x_j \in S\{\lambda_j\}$ and $x_i \lessdot x_j$, then $\lambda_i \lessdot \lambda_j$.

As has been discussed before, the concept of direct transition between subsets of operations is important for the establishment of linear sequences without violating precedence relationships. On the other hand for the establishment of compound sequences, the concept of direct transition between subsets of operations comes into question. For the former, c(i,j) represents conditions of direct transition, while for the latter, $p\{\lambda\}$. Taking these into consideration and arranging the process for establishing a new compound sequence ${}^{m}\xi_{i'}(\lambda_1, \lambda_m)$ from ${}^{m-1}\xi_i(\lambda_1, \lambda_{m-1})$ and λ_m without violating precedence relationships, results in the definition of subsequential multiplication whose flow chart is shown in

Fig. 3.14.

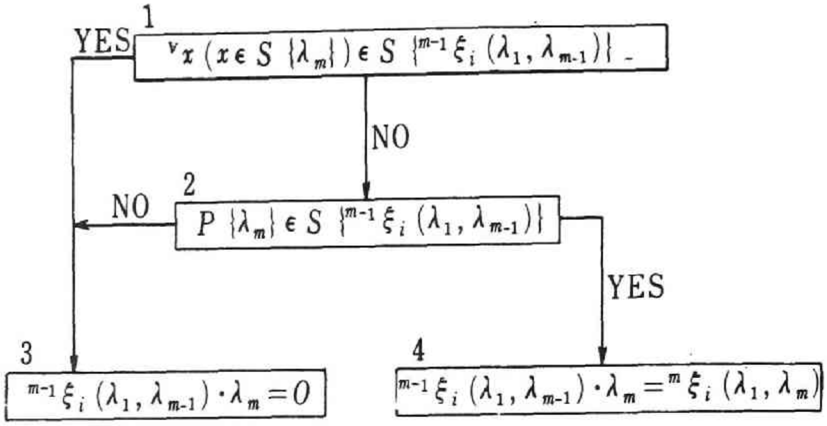


Fig.3.14. Subsequential multiplication.

Box 1 checks whether $\forall_x$ in $S\{\lambda_m\}$ is contained in $S\{^{m-1}\xi_i(\lambda_1, \lambda_m\}$. Box 2 checks whether or not the precedence operations, $P\{\lambda_m\}$, are included in $S\{^{m-1}\xi_i(\lambda_1, \lambda_m)\}$. If the answers of Box 1 and Box 2 are "YES," then a new compound sequence is constructed.

The establishment of compound sequences is carried out in the following manner with subsequential multiplication. The subsets of operations which require no precedence operations are picked up first. They are denoted by $X(1)$. Then the subsets of operations which require no precedence operations except $X(1)$ are listed to construct compound sequences $\{X(1), X(2)\}$. The step is carried out successively. If $|X| = n$, all of the compound sequences can be obtained at most by n steps.

3. 5 CONCLUSIONS

This chapter introduced a method of establishing compound sequences. Compound sequences were considered by first constructing feasible subsets of operations, and this approach was found to be advantageous from the point of view of being able to handle various technological restrictions and conditions otherwise difficult to formulate, to say nothing of precedence relationships. It is also meritorious in that appreciable reduction in calculation times necessary for determining an optimum sequence was possible. From the viewpoint of the above method, subsequential product was introduced for constructing a feasible subset of operations, and to prevent the overlapping of a set of these subsets the combinatorial matrix approach was tried. Moreover consideration of the problem of the minimum-time array for overlap type compound sequences resulted in the formulation of several effective theorems. Then subsequential multiplication of subsets of operations was introduced for the establishment of compound sequences. This leads to easy, systematic establishment of compound sequences.

CHAPTER 4 DECISIONS OF OPTIMUM COMPOUND SEQUENCES

4. 1 TWO PROBLEMS

This chapter deals with the following two problems by the method of establishing compound sequences developed in Chapter 3.

(1) The problem of determining an optimum compound sequence which is composed of subsets of operations to minimize the sum of subset values associated with them having precedence restrictions.

(2) The line balancing problem.

The first problem is going to be formulated to combine desirable subsets of operations into a compound sequence to lend itself to designing transfer machines, etc. In view of one or more specified criteria, some value, termed subset value, is given to each of subsets of operations. It is expected to minimize the sum of subset values associated with subsets of operations which construct a compound sequence. The algorithm similar to the one developed for the problem of minimizing the sum of sequence-dependent setup-times with precedence restrictions is to be introduced for the first problem in Section 4.2.

The second problem exists where a number of operations must be performed sequentially with certain constraints.

These constraints are concerned with the ordering of operations, the organization of the line, and the required production rate. Given an output rate to flow from the line, how can the operations be grouped and wholly assigned to stations so that a minimum number of stations are required ? By introducing the concept of lower bound on idle time the problem is tackled in Section 4.3. The goal is first to develop an optimum procedure for the assignment, and them for large scale problems to develop an acceptable procedure using certain new problem-solving techniques.

4. 2 DECISION OF AN OPTIMUM COMPOUND SEQUENCE TO MINIMIZE THE SUM OF SUBSET VALUES WITH PRECEDENCE RESTRICTIONS

4.2.1 Problem Statement

The operations to be performed are identified by the integers $X = \{1, 2, \ldots, n\}$. Subsets of operations are constructed from the set X . It is desired to combine desirable subsets of operations into a compound sequence. What subsets of operations should be used to complete the operations ? One or more specified criteria to evaluate subsets of operations could be combined into one criterion by giving weight to each of the criteria according to their importance. The resultant criterion is to be optimized. The value which is given to a subset of operations λ_i is called a subset value, and is denoted by $r(\lambda_i)$. Therefore, the problem is to find, for a set of subsets of operations constructed from a finite set X, a collection of subsets of operations of X, satisfying the following three conditions :

(1) $\bigcup_i \lambda_i = X$

(2) $\lambda_i \cap \lambda_j = \phi \qquad (i \neq j)$

(3) If $x \in S\{\lambda_i\}$, $y \in S\{\lambda_i\}$ and $x \leqslant y$, then $\lambda_i \leqslant \lambda_j$,

and minimizing the following function :

(4) Min $\sum_i r(\lambda_i)$.

4.2.2. Algorithm

A branch and bound approach is used in this section for the reasons that it can guarantee optimality, seems reasonable to program and is generally applicable. Furthermore, the Algorithm 1 developed in Section 2.2 for the problem of sequencing a set of jobs to minimize the sum of sequence-dependent setup-times with precedence restrictions can, fortunately, be applied with amendments. Necessary amendments are shown in the following :

• Linear sequence ${}^m\pi_\nu(x_1, x_1)$	• Compound sequence ${}^m\xi_\nu(\lambda_1, \lambda_m)$
• Setup-time $s(x_i, x_j)$	• Subset value $r(\lambda_i)$
• Fundamental matrix with setup-times	• Table of subsets of operations with subset values
• Reduction of matrices	• Reduction of tables

- Crossing off row x_i and column x_j after selecting (x_i, x_j)
- Crossing off subsets of operations which include some of x_i, x_j, ... , x_k, after selecting subset of operations $\{x_i, x_j, \ldots x_k\}$

Reduction of matrices by assignment model optimization is not applied to the problem.

To avoid redundancy, the amended algorithm is illustrated by a numerical example. For the precedence diagram shown in Fig. 1.1, suppose that a table of subsets of operations is given as in Table 4.1. There might be more subsets of operations but anyhow suppose that feasible subsets of operations are only those listed in the table. The first column shows subsets of operations : the second, subset values ; the third, reduced values ; the fourth, $p\{\lambda_i\}$. Reduced values change according to the order of reducing. The reduced values in the table were obtained reducing in the order of operations 1, 2, ... , 10. The total sum of reducing constants is $l(\Gamma_1) = 283$. After this problem statement, the amended algorithm is applied to the problem. The result is shown in Fig. 4.1. From the table, the subsets of operations which can be performed first are (1), (1,2), and (1,3) .

As to (1), the subsets of operations which include (1) are crossed off from the table and it is reduced with regard to the resultant operations 2,3, ... , 10 in such a way that each operation can be processed with a zero reduced value. The result is shown in Table 4.2(a). As to (1.2) and (1.3) reduction is done in the same manner. The result of case (1.2) is shown in Table 4.2(b). Since subset of operations (1.3) has the smallest lower bound, it is going to be branched further. Carrying out branching well enough to get an optimum sequence results in Fig. 4.1. The optimum sequence is $\xi = (1\cdot3)(5)(2\cdot6)(8)(4\cdot7)(9\cdot10)$.

Checking optimality is much easier in this case than that of Algorithm 1. In Fig. 4.1, the knots which have an asterisk (*)

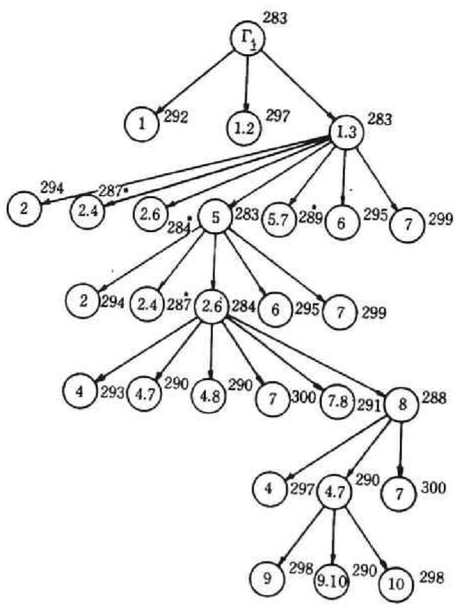


Fig. 4.1. Tree-like diagram for the problem.

Table 4.1. Table of subsets of operations

λi	subset values	reduced values	$P\{\ \lambda i\ \}$
1	38	0	
1.2	61	0	
1.3	72	0	
2	29	6	1
2.4	63	0	1
2.6	58	0	3
3	43	9	1
3.4	90	16	2
3.7	76	14	1
4	50	10	2
4.7	69	1	2, 3
4.8	62	3	2, 5, 6
5	18	0	3
5.7	46	0	3
5.9	56	13	4, 6, 7
6	47	12	3
6.9	64	4	4, 5, 7
7	38	10	3
7.8	48	1	2, 5, 6
7.9	53	0	4, 5, 6
8	19	0	2, 5, 6
8.9	52	8	4, 5, 6, 7
8.10	42	0	9
9	39	14	4, 5, 6, 7
9.10	54	6	4, 7, 8
10	27	4	9

Table 4.2. Reduced tables

λi	reduced values
2	6
2.4	0
2.6	0
3	0 *
3.4	7 *
3.7	5 *
4	10
4.7	1
4.8	3
5	0
5.7	0
5.9	13
6	12
6.9	4
7	10
7.8	1
7.9	0
8	0
8.9	8
8.10	0
9	14
9.10	6
10	4

(a)

λi	reduced values
3	0 *
3.4	7 *
3.7	5 *
4	9 *
4.7	0 *
4.8	2 *
5	0
5.7	0
5.9	13
6	8 *
6.9	0 *
9	10
7.8	1
7.9	0
8	0
8.9	8
8.10	0
9	14
9.10	6
10	4

(b)

by lower bounds has smaller lower bounds than the lower bound of an optimum sequence. Branching these knots further is omitted for avoiding complexity since the order of subsets of operations in a compound sequence can be changed as far as given precedence relationships are not violated.

A large scale problem might require compromise on an approximate solution which can be obtained by branching only in one direction and omitting to check optimality. The Algorithm 1 can extensively be used for the problem. It is quite interesting that different problems can be solved by the same way of thinking, which indicates that the way of thinking is quite useful.

4.3 THE LINE BALANCING PROBLEM

4.3.1 Problem Statement

The manufacturing of a certain commodity is accomplished by the use of a series production lines. The basic components of such a line are n indivisible, elemental operations $X = \{1, 2, \ldots, n\}$ for which the processing-times are assumed to be known constants. These times are also assumed to be independent of the sequence in which the operations are performed. The technology of such processes usually imposes a set of constraints, called precedence relationships and zoing restrictions and so forth, on these operations. Zoing restrictions occur when certain operations are physically incompatible and can not be assigned to the same station. A series production line is designed by aggregating elemental operations at work stations in such a way that these constraints are not violated. The time required by the station taking the longest to complete the operations assigned to it is called the cycle time of the line and is denoted by C.

Formally, the problem is to find, for a set of subsets of elemental operations constructed from a finite set X, a collection of subsets of operations of X, satisfying the following four conditions :

(1) $\bigcup_i \lambda_i = X$,

(2) $\lambda_i \cap \lambda_j = \phi \qquad (i \neq j)$,

(3) If $x \in S\{\lambda_i\}$, $y \in S\{\lambda_j\}$ and if $x \prec y$, then $\lambda_i \leqq \lambda_j$,

(4) $t(\lambda_i) \leqq C$.

and optimizing some criterion.

The purpose of determining a sequence of a line is to expect to minimize the total cost necessary per unit. The cost items which affect the total cost include direct labor cost, inventory cost, facility cost, breakdown cost, etc. A number of approaches [1)~18)] have been made to finding a solution that minimizes the total number of work stations in expectation of minimizing direct labor cost. Although the objective function brings out a question, it is of practical interest since there is no explicit criterion which includes those cost items simultaneously so as to satisfy the aim of determining a sequence of the line. As it is not necessarily the only one that minimizes the total number of work stations, those cost items could be considered after as many sequences as necessary which minimize the total number of work stations have been established.

From this discussion, in what follows an algorithm which raises the efficiency of the line by minimizing the total idle time will be established.

The difference between the time required by any station to complete its operations and the cycle time is called the idle time of the station. It is conventional to take the sum of all station idle times (called total idle time) as a measure of the efficiency of the design of a line. A subset of operations which is performed at the i th station is denoted by $\lambda_i^{p_i}$, where p_i represents the dimension of the array and is omitted if not necessary.

- Processing-time of $\lambda_i^{p_i}$: $\tau(\lambda_i^{p_i}) = \sum_{k \in \lambda_i^{p_i}} t(x_k)$ (4.1)
- Idle time of $\lambda_i^{p_i}$: $d(\lambda_i^{p_i}) = p_i \cdot C - \tau(\lambda_i^{p_i})$ (4.2)

The efficiency of the i th station E_i is as follows :

$$\cdot \quad E_i = \frac{\tau(\lambda_i^{p_i})}{p_i \cdot C} \times 100 \qquad (\%) \qquad (4.3)$$

The efficiency of the line E is,

$$E = \frac{\Sigma E_i}{N} \quad , \qquad (4.4)$$

assuming that the total number of work stations is N.

Total idle time d is

$$d = \sum_i d(\lambda_i^{p_i}) \qquad (4.5)$$

Since $\sum_i t(x_i)$ is constant, to maximize the line efficiency is equivalent to minimizing $(p_1+p_2+ \dots +p_i) \cdot C$ or total idle time d.

In what follows, the algorithm will be established to maximize the line efficiency (4.4) first and them to minimize N so as to make the line as compact as possible. Since there are conceivably more than one combination which minimize N, to minimize N itself is not a good criterion. From a practical point of view, the array dimensions of the work stations are desired to be the same as far as a good balance is obtained. Otherwise the array dimension of the work station which has a lower efficiency should be decreased. So the problem is to determine array dimension, which is determined by technological and economic restrictions. Therefore the main problem is to minimize the total idle time.

4.3.2 The Algorithm

The most proper approach to be taken in the first place, in order to solve a combinatorial problem, if a special effective method can not be found, is to try to reduce by some methods the existential range of an optimum solution. For the purpose the following fact is effectively used :

In case there are more than one subset of operations which can be assigned to a work station, if it can be concluded , without going any further, that subset λ_i must be at least as good as λ_j , for the reason that if λ_j yields the minimum number of work stations, and that λ_i must do so also, then λ_j need not be considered. The endeavor to make use of the fact results in the following theorem.

Theorem 4.1 Suppose that an optimum partial sequence ${}^{i-1}\xi(\lambda_1, \lambda_{i-1})$ whose subsets of operations are $\lambda_1, \lambda_2, \cdots,$ λ_{i-1} has been obtained already and that subsets operations λ_{i_1} and λ_{i_2} can be assigned to the i th station. Let operations which are not common to λ_{i_1} and λ_{i_2} be $x_{11}, x_{12}, \cdots,$ and x_{1p} ; and $x_{21}, x_{22}, \cdots,$ and $x_{2q},$ respectively. Suppose further that there exist subsets of operations which do not include $x_{21}, x_{22}, \cdots,$ and x_{2q} at all, but include some or all of $x_{11}, x_{12}, \cdots,$ and x_{1p} . Let λ_A be the set of such subsets of operations which wholly include all of the operations $x_{11}, x_{12}, \cdots,$ and x_{1p}. For possible

λ_A exchanging the operations $x_{11}, x_{12}, \cdots,$ and x_{1p} in λ_A and the operations $x_{21}, x_{22}, \cdots,$ and x_{2q}, results in

(1) a set of feasible subsets of operations,

and

(2) $F\{x_{21}, x_{22}, \cdots, x_{2q}\} \in \{x_{11}, x_{12}, \cdots, x_{1p}\} \cup F_A\{x_{11}, x_{12}, \cdots, x_{1p}\}$,

then λ_{i_1} should be assigned to the i th station.

(Proof) Suppose that a complete compound sequence has been obtained by assigning λ_{i_2} to the i th station. Then the operations $x_{11}, x_{12}, \cdots, x_{1p}$ and $F_A\{x_{11}, x_{12}, \cdots, x_{1p}\}$ have been assigned to the stations after the i th station. From the conditions (1) and (2), the operations $x_{21}, \cdots, x_{2q}$ and the operations $x_{11}, \cdots, x_{1p}$ can be exchanged without violatin required precedence relationships and exceeding given cycle time. That is, the sequence which assigns λ_{i_1} to the i th station can be always obtained from the sequence which assigns λ_{i_2} to the same station. But the converse is not always true. After all, if λ_{i_1} and λ_{i_2} which satisfy the above conditions (1) and (2) can be assigned to the i th station, λ_{i_1} should always be selected.

Notice that in this theorem the restriction on time is not being expressed explicitly. The two rules, dominance and duplication, which Jackson[1] pointed out in his paper are the special cases of this theorem.

For the example whose precedence diagram is shown in

Fig. 4.2, where numbers outside the circles represent processing-times, suppose that C = 19 and that sequence is required to be of linear type. A subset of operations (1.4) is chosen as a set of the operations assigned to the first station. Then 75 subsets of operations can be considered as a set of the operations assigned to the second station. An application of the theorem results in 10 subsets of operations as shown in the second steps of Fig. 4.3.

By the theorem the existential range of possible assignments to stations can be reduced, but it is quite usual that for practical large scale problem the number of possible subsets to be assigned to stations become intractable. Therefore, consider the merits and demerits of selecting a certain subset of operations among subsets of operations as a candidate to be assigned to the i th station. This suggests investigating the influence of the selected subset on performing the unfinished operations, in reference to the other candidates. The previous theorem offers quite a useful and effective means to cross off unnecessary candidates, but has a limitation that it does not make use of the information effectively which the unfinished operations have. Ranked positional weight technique by Helgeson and Birnie[2)] is one of the endeavors to make use of the information. But the technique has, unfortunately, not succeeded in making

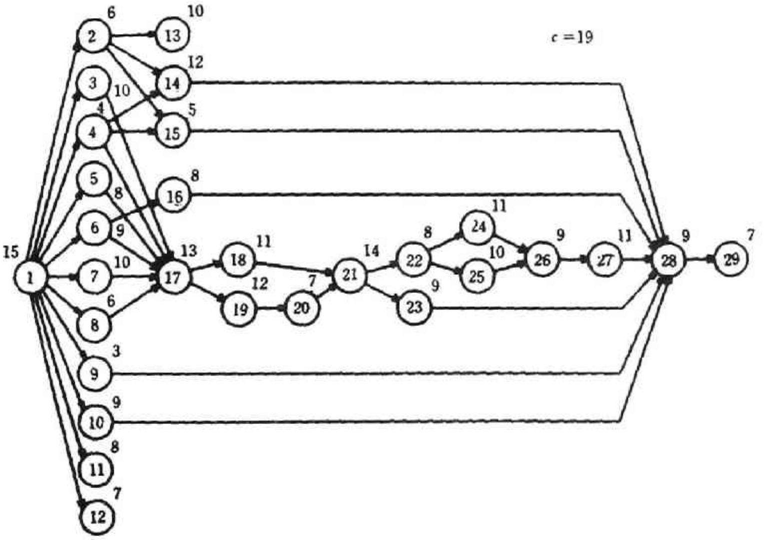


Fig. 4.2. Numerical example.

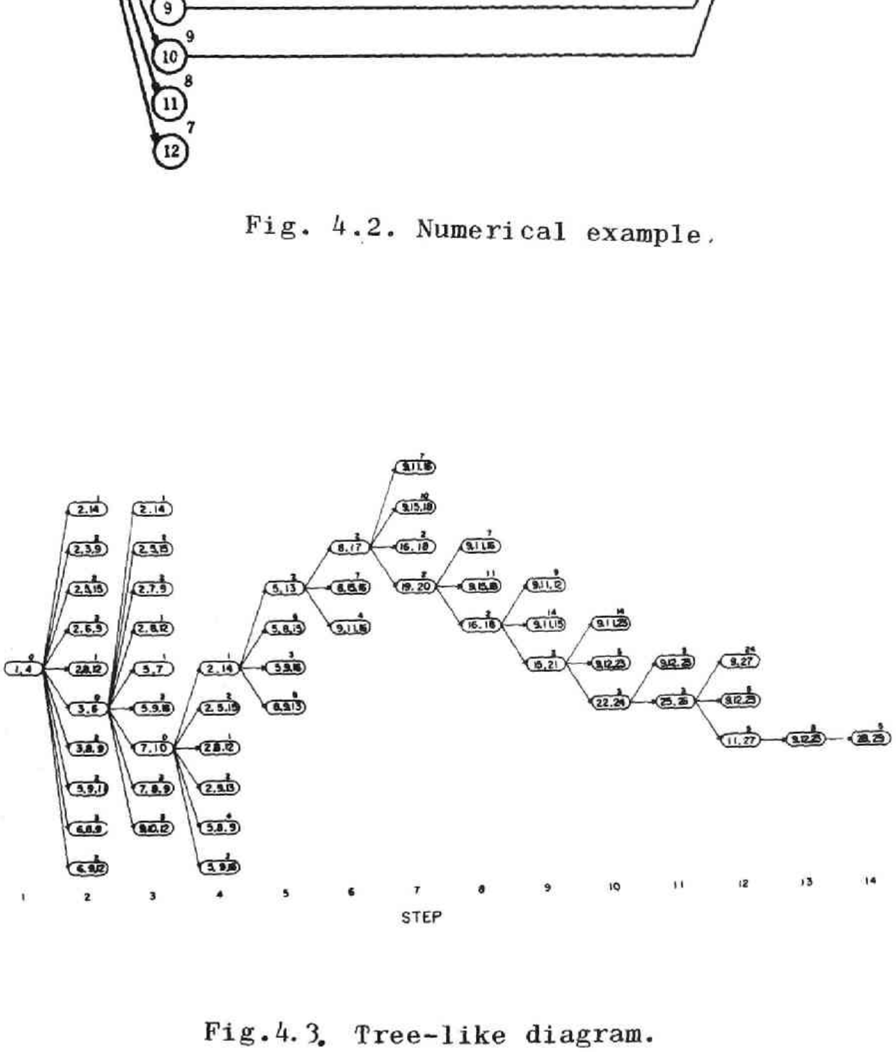


Fig.4.3. Tree-like diagram.

use of the information fully.

One method which utilizes the information almost perfectly is to calculate the sum of the idle time which each unfinished operation requires at least. For this purpose the concept of lower bound on idle time is introduced.

Suppose that a partial sequence ${}^{i}\xi(\lambda_1, \lambda_i)$ has been obtained. The lower bound on idle time which $d\{{}^{i}\xi(\lambda_1, \lambda_i)\}$ has is d' plus d'', where,

d' is the sum of idle times of ${}^{i}\xi(\lambda_1, \lambda_i)$,

d'' is the sum of idle times at least necessary to assign the unfinished operations to stations thereafter,

$$d\{{}^{i}\xi(\lambda_1, \lambda_i)\} = d' + d'' .$$

The value d'' of the selected sequence can be obtained by calculating for each unfinished operation the least idle time among subsets of operations which include the operation and which does not include any of the operations included already in $\lambda_1, \lambda_2, \cdots,$ and λ_i . On this occasion the following notice is necessary to avoid concurrence of summing the least idle times. That is, if some unfinished operation included in a certain subset of operations requires a positive idle time, then the idle times of the other unfinished operations in the subset should not be calculated any more. This consideration might be ambiguous, but it is adequate for calculating lower bound on time. For the example, Fig. 4.4 shows the result of the combinatorial analysis of the resultant 10 subsets of operations in the second step of Fig. 4.3.

As there are conceivably more than one subset to be assigned next to λ_i , the lower bound for each of candidates should be calculated in this manner. As the next assignment λ_{i+1} the subset of operations which has the minimum lower bound among these is selected.

quence	candidates	d'	2	3	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	d''	d
	(2,14)	1	X											X																0	1
	(2,3,9)	0	X	X					X																				2	2	2
	(2,5,15)	0	X		X										X						2									2	2
	(2,6,9)	1	X			X			X																				2	2	3
,4)	(2,8,12)	0	X					X				X			✓		1													0	1
	(3,6)	0		X		X																								0	0
	(3,8,9)	0		X				X	X																				2	2	2
	(5,9,11)	0			X				X		X																		2	2	2
	(6,8,9)	1				X		X	X																				2	2	3
	(6,9,12)	0				X			X			X																	2	2	2

Fig. 4.4. Combinatorial analysis.

Selecting it at this step because it has the highest possibility to yield the minimum total idle time does not assure optimality. Therefore when total idle time exceeds an allowable value afterwards, return to this step and reselect another subset of operations as the next assignment. In some cases there exist still several subsets of operations which have the minimum lower bound. In such cases, select one anyway preferably the one composed of a fewer operations with strict precedence relationships.

For the previous example, subset of operations (3.6) is selected as the assignment to the second station.

(1) Flow chart of the algorithm.

From the above analysis an algorithm can be developed. The flow chart of linear type is shown in Fig. 4.5. Block 1 constructs a set of feasible subsets of operations for a given precedence diagram and the cycle time. Calculate

$$d_{opt} = N_{nes} \cdot C - \sum_{i=1}^{n} t(x_i), \tag{4.7}$$

where d_{opt} is the total idle time allowed to an optimum sequence, N_{nes} is the number of stations necessary to complete X ;

$$N_{nes} \geqq \frac{\sum_{i} t(x_i)}{C} \tag{4.8}$$

N_{nes} can be determined from the desirable efficiency of the line[3]. In case of overlap type, instead of (4.7),

$$d_{opt} = (p_1 + p_2 + \cdots + p_{N_{nes}}) \cdot C - \sum_{i=1}^{n} t(x_i) \tag{4.9}$$

For the example, $d_{opt} = 5$. Block 2 lists the subsets of operations to be assigned to the first station and arrives at List 1-A by crossing off unnecessary subsets of operations by the theorem. Block 3 selects as the assignment to the first station the subsets of operations which give the minimum idle time in case List 1-A has more than one subset of operations. Block 4 checks if there are more than one sub-set of operations which give the minimum idle time. If the answer is YES, go to Block 5 and select only one subset of operations. Block 6 checks if the idle time which the selected partial sequence has is less than or equal to d_{opt}. If not, Block 7 sets

$$d_{opt} = d_{opt} + C \quad .$$

Block 8 to Block 13 makes a loop. Passing through the loop successively proceeds branching processes. Block 8,9,10 and 11 correspond to Block 2,3,4, and 5, respectively. They are also the same for overlap type. In case the sum of idle times d exceeds the d_{opt} , the dimension should be decreased by one. Block 12 checks if the time which the selected partial sequence ${}^{n+1}\xi(\lambda_1, \lambda_{n+1})$ is less than or equal to d_{opt} . If the answer is YES, go to Block 13, which checks the completeness of the selected sequence. If the answer is NO, go to Block 8 and make further branching. If the answer is YES, stop, since the selected sequence is an optimum sequence. In Block 12, if the idle time of the selected sequence exceeeds d_{opt} , go to Block 14, which looks for an unbranched sequence which has the sum of idle times which is less than or equal to d_{opt} . If there exists such a sequence, select in Block 16, among sequences which have the sum of idle times less than or equal to d_{opt} and which have branched farthest, a sequence which has the minimum idle time. In case of overlap type, select a sequence which has the minimum idle time among sequences which have the highest value of $(p_1 + \dots + p_i)$ and which have idle time less than or equal to d_{opt} . In Block 14, if there

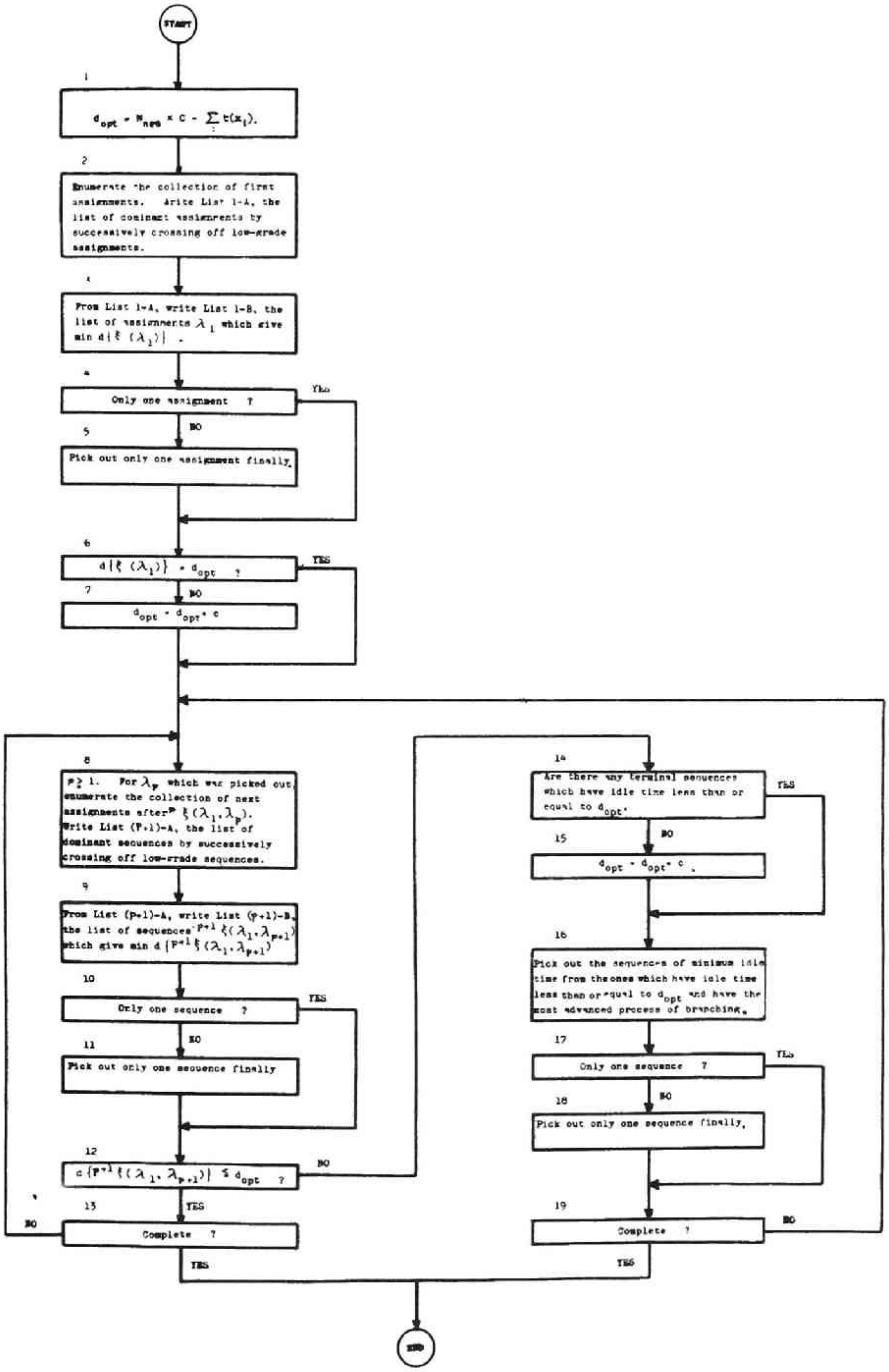


Fig. 4.5. Flow chart.

exists no sequence which has an idle time less than or equal to d_{opt}, set $d_{opt} = d_{opt} + C$ in Block 15. Block 16 checks if there are more than one sequence which gives the minimum idle time. If the answer is zero, go to Block 17, which selects only one sequence. Block 19 checks if the selected sequence either in Block 17 or in Block 18 satisfies the completeness. If the answer is NO, return to Block 8. If the answer is YES, stop, since the solution is optimum.

An application of the algorithm to the previous example shown in Fig. 4.2, results in the tree-like diagram shown in Fig. 4.3. The subsets of operations shown at each step are the ones which have not been crossed off by the theorem. The numbers outside the boxes show the sum of idle times. Block 14 to Block 18 were not used for this example.

(2) Characteristics of the algorithm.

The characteristics of the algorithm are in elimination of unnecessary inferior subsets of operations and utilization of the concept of lower bound on idle time. These can reduce the number of trial and error to find an optimum sequence and make up for the lack of criterion of assigning subsets of operations to stations. The algorithm can assure optimality and find alternative sequences if necessary.

4.3.3 Discussions

(1) Evaluation of total idle time .

The problem of applying the algorithm is how to determine proper cycle time and evaluate total idle time. The larger the scale of a problem becomes, the more total idle time should be assumed.

If an optimum sequence will be determined after constructing all of the feasible subsets of operations, then the total number of all feasible subsets of operations is another problem. The total number of all feasible subsets of operations depends on the number of operations, the processing-times, given cycle times, required precedence relationships and so on. The total number of all feasible subsets of operations for the previous example is 331 [Note : The example has about 1,800,000,000,000 linear sequences as discussed in Chapter 1] . There is no problem in the case of such a small-scale problem, but as is easily understood, the computation becomes intractable for large scale problems. The algorithm does not necessarily require that all of feasible subsets of operations be constructed. It is sufficient to make only desirable feasible subsets of operations if they cover all of the operations to be performed, and the algorithm can find an optimum sequence among them.

(2) Introduction of a time chart.

It is necessary to evaluate total idle time effectively for a large-scale problem. One way of doing this is to perform the following operation when the theorem can not be applied : Note that each operation included in a candidate to be assigned to a station has limited influence on the unfinished operation from the viewpoint of line balancing. The solution is to restrict the calculation of idle time to the unfinished operations under the influence. Of course as estimation of idle time becomes approximate, the convergence of the algorithm will become slow.

It is convenient to introduce a time chart to make a rough estimation of idle times of unfinished operations. In the time chart, the X-axis is taken in the direction of the time and is divided by cycle time C. Each operation is expressed in the time chart according to its earliest starting time. For example, Fig. 4.6 shows a time chart after selecting partial sequence $\pi = \{(1,4), (3,6)\}$. In the figure, note that operation 17 is expressed after 6 in the second division since the precedence operations are 5,7, and 8 and balancing 5,7, and 8 anyhow results in expressing operation 17 at least after 6 in the second division, but as to operation 28, the calculation of the earliest starting time is difficult. To take an easy way of calculation is to divide the sum of the processing-times of the precedence operations by C. This method is not accurate but the utilization of a time chart itself is just an easy way of making an estimation of total idle time, and therefore the above compromise should be admitted.

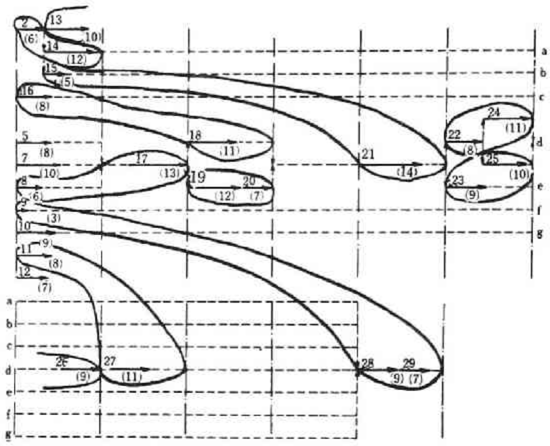


Fig. 4.6. Time chart after selecting (1,4), (3.6).

Estimation of total idle time for a complicated precedence diagram as in Fig. 4.7 is much more difficult.

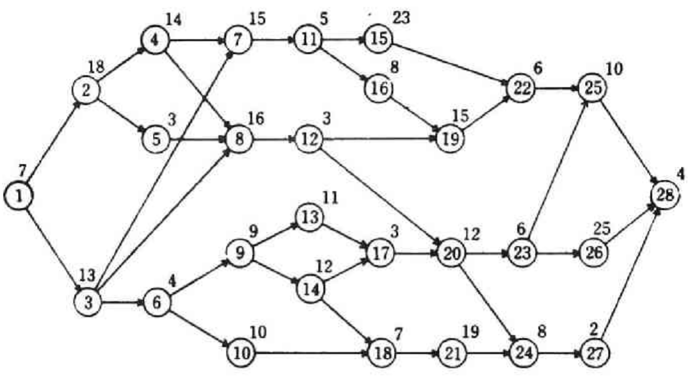


Fig. 4.7. Precedence diagram. An optimum compound sequence of overlap type of two dimensions is sought.

In such a case dividing the diagram by cycle time C should satisfy. Fig. 4.8.(a), and (6) are such examples.

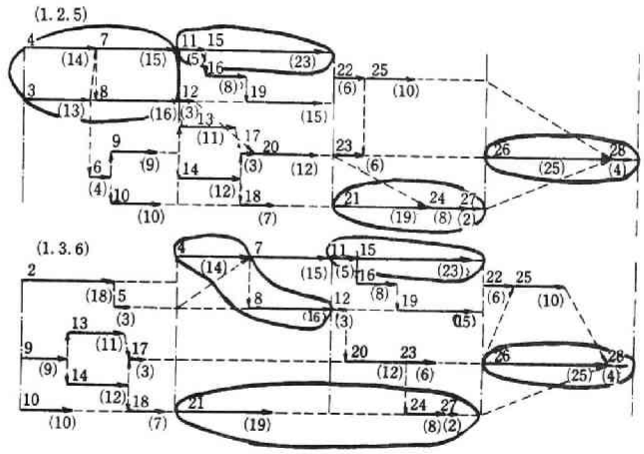


Fig. 4.8. Division of the precedence diagram shown in Fig.4.7 by C after selecting either (1,2,5) or (1,3,6).

The first assignment in (a) is (1,2,5), and the second is (1,3,6). Thereafter compound sequences of two dimensions are sought. In the time charts attention should be paid to the operations which are extending the time axis farthest. The way of thinking is in some ways similar to a critical path in a network diagram. But the time chart is divided by cycle time C into several portions. Therefore the set of operations which extend the time axis farthest may be different from the set of operations which form a critical path. In case operations in each divided portion have a good balance, they are deleted, otherwise they are combined with operations in other portions so as to have a better balance. Fig. 4.6 and Fig. 4.8.(a) and (b) show manner in which this is done. In Fig. 4.6, an optimum solution can, fortunately, be found quite easily by looking at it and performing calculations by trial and error. In Fig. 8 (a), and (b) more calculation is necessary to know which of subsets of operations (1,2,5) and (1,3,6) yields a better balance. For the purpose, cross off the operations which have good balance from the chart and check whether or not the resultant operations can have good balance. For the example, case (a) yields a good balance but (b) does not.

If both cases (a) and (b) do not yield good balance, either (1) allowable total idle time should be altered, or (2) the way of combining unfinished operations into good balance has been inefficient. For such a case, branch farther and decide which subset of operations should be taken.

For the example, subset of operations (1,2,5) is assigned to the first station since case (a) yields good balance. Continuing the necessary branching to obtain an optimum sequence results in the tree-like diagram shown in Fig. 4.9.

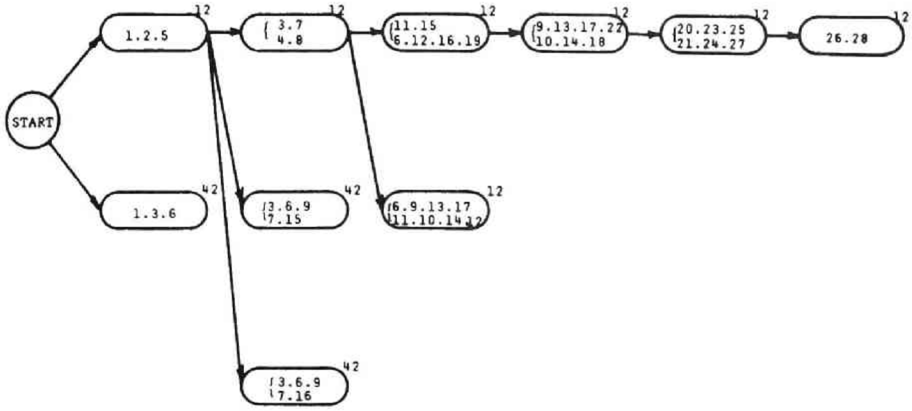


Fig. 4.9. Tree diagram. The numbers outside boxes show upper bound on total idle time.

The optimum solution is

$$\xi = (1,\ 2,\ 5)\left(\begin{cases}3\\4\end{cases}\middle|\begin{matrix}7\\8\end{matrix}\right)\left(\begin{cases}11,\ 15\\6,\ 12,\ 16,\ 19\end{cases}\right)\left(\begin{cases}9,\ 13,\ 17,\ 22\\10,\ 14,\ 18\end{cases}\right)\left(\begin{cases}20,\ 23,\ 25\\21,\ 24,\ 27\end{cases}\right)(26,\ 28)$$

By the method of introducing a time chart, upper bound on total idle time is looked for, that is, minimax strategy is taken since a sequence which minimizes upper bound on total idle time is sought. Therefore in case upper bound is equal to d_{opt}, then there is no problem, but if it exceeds d_{opt}, the checking of optimality becomes rather difficult. In the previous algorithm if it goes into box 15, that is, if d_{opt} which was first set is not suitable and must be altered, then the checking of optimality requires extraordinary pains. Therefore, for the previous algorithm it is essential to set proper d_{opt}. By introducing the time chart, d_{opt} can be roughly estimated quite easily.

4. 4 CONCLUSIONS

This chapter dealt with two problems which are encountered often in production lines : The problem of minimizing the sum of subset values with precedence restrictions, and the line balancing problem.

The algorithm similar to the one developed for the problem of minimizing the sum of setup-times with precedence restrictions has been proposed for the first problem.

The algorithm of the line balancing problem has been developed based on elimination of unnecessary inferior subsets of operations and utilization of the concept of lower on idle time.

By combining the two algorithms a more general line balancing problem to minimize the total cost necessary for manufacturing one commodity unit could be solved.

PART II

IN-PROCESS INVENTORY CONTROL

OF

PRODUCTION LINES

INTRODUCTION

The utilization of production lines associated with high volume production is a common characteristic of modern machine industry. The production line provides less flexibility than the job shop ; however its advantages are less material handling, improved man and equipment utilization, reduction of inventory and time in process, flow-space saving, ease and simplification of production control, etc. These advantages all help to lower production costs.

The complete specification of a production line design contains a rather large number of decisions and considerations. The production line consists of a number of interconnected stations or stages (a stage : a group of stations) at which operations are performed on workpieces in order to convert the inputs to the system into outputs of the system. The operations in the system are performed by some equipment which is liable to failure or breakdown. Breakdowns must be repaired and production from the station is lost during repairs. By linking the stages to form a line, the efficiency of it is decreased significantly compared to the use of an individual machine with the consequence that if any one station stops, all other stations in the line are forced to shut down. One way of improving the line efficiency is to provide buffer stocks between certain stages or sections of the line. A storage facility between two successive stages is called a buffer. A group of stations located between two in-process inventory banks, but having no in-process inventory storage within the group is regarded as a stage. The in-process buffers decouple the production stages and diminish the forced down effect caused by stage breakdown. Such buffer storage occupies valuable space, the workpieces kept in it has high storage cost and associated with handling the unit into and out of these in-process inventory banks is the storage facility cost. The purpose of this PART II is to give better guidance on how much inter-stage storage capacity should be provided, what the effect of given buffer capacity on the line efficiency is, how the stages should be placed, and how to allocate the storage capacity among the stages :

Each of the above decisions is subject to technical and economic constraints. Within them all of these decisions should be made so as to maximize the profit margin realized from the line or minimize the capital expenditure and operating costs of the production line.

It will be assumed throughout this PART II that in-process inventory storage can be provided in between some or all of the stations of the production line. In general, in-process inventory storage has limited capacity for operational and economic reasons. It will be assumed further that the system is processing only one commodity.

The purpose of Chapter 5 is to gain insight into the problem on the role of buffer stocks in production lines and present the results of a theoretical study of the problem. After reviewing earlier theoretical work on the problem, the formulation of the problem is given, and then a Markovian process model of production buffer systems is proposed. It is shown that the model is a useful tool to analyze the role of in-process inventory banks in production lines in improving the line efficiency.

A two stage line consists of two groups of stations separated by a buffer. The reasons for installing buffers can be illustrated most simply by considering two stages. Based on the analysis developed in Chapter 5, for two stage lines Chapter 6 presents answers to the following :

(1) Should buffer stocks be used ?

(2) What is the effect of given buffer capacity on the line efficiency, or on the number of mean buffer stocks ?

(3) What is the effect of variation of breakdown rates, repair rates, or stage efficiencies ?

After answering these questions, cost analysis of a production line is made to help the system designer make a better decision so as to maximize the system profit.

From a computational point of view, three and more stage models tend to be intractable. This observation demands other problem-solving procedures, one of which is computer simulation. The obvious limitation of simulation is that it does not give mathematically proven solutions. However interpretation of simulation results with the aid of the results obtained in the previous chapters can provide significant insight into the behavior of multi-stage lines with buffers. Chapter 7 presents a very useful computer simulation model to investigate the behavior of production lines having any number of production stages, any size buffer inventory, and any breakdown time and repair time distributions at any stage.

Based on the simulation model developed in Chapter 7, Chapter 8 provides answers for mainly the following questions regarding multi-stage lines :

(1) How are the effects of the number of stages on the relationship between the line efficiency and buffer capacity ?

(2) How should given buffer capacity be allocated among the stages ?

(3) In which order should the stages be placed ?

Thus, this PART II gives an exposition of in-process inventory control of production lines .

CHAPTER 5 ANALYSIS OF THE EFFECTS OF BUFFER STORAGE CAPACITY

5. 1 INTRODUCTION

The purpose of this chapter is to gain insight into the problem on the role of buffer stocks in production lines and present the results of a theoretical study of the problem.

A Markovian process model of production buffer systems is proposed, and it is shown that it is a useful tool to analyze the role of in-process inventory banks in production lines in improving the line efficiency.

As to previous work on the problem, the work in queueing theory by Hunt[1] should be mentioned. He provided the initial studies on the behavior of the production line and analyzed the maximum utilization and expected number of units of a two-stage model under different assumptions with respect to the storage size between the stages. For a review of earlier theoretical work on the problem see Koenigsberg[2]. He quotes an unpublished study of the two stage line by Finch, who assumes that the breakdown rate of a forced down stage equals the breakdown rate of an operating stage. This does not seem to be a reasonable assumption. Others who have worked on the problem analytically are Morse[3], Elmaghraby[4], Buchan and Koenigsberg[5], Patterson[6], and Hillier and Boling[7), 8)]. These works have utilized for computational tractability the Poisson arrival and exponential service time assumption which does not represent, the majority of actual automated production lines, and the results are, in general, limited to small production lines. Love[9)] analyzed a two stage line by coupling two backlogging models together. For the three stage model with exponential service time, Hatcher[10)] has developed the closed-form expressions for the steady state probabilities. Models for more than three stages have caused considerable trouble since the effects of blocking of the system when one or more of the storages are full are difficult to formulate. Buzacott[11)] discussed the general effect of in-process inventory for automatic transfer lines. For related studies see Buzacott[12), 13)] and Knott[14)]. Kay[15)] has described a case study of an automated bottling plant in a brewery.

5. 2 PROBLEM STATEMENT

While production lines can take various forms, the following presents a picture of actual production lines : The line produces one kind of commodity, consisting of a number of stages at each of which an operation is carried out on a workpiece. The stages are arranged serially so that each workpiece enters the line at the same stage and transfers from one stage to the next till it has passed through the final stage. All workpieces begin to transfer from one stage to the next at the same instant. The interval between successive transfers is called the cycle time. There is always a supply of workpieces available to the first stage of the production line. The final stage will deposit the completed workpiece into a storage area which has an infinite capacity. Each storage point has a fixed capacity ; the capacity of the storage point between stage i and stage i+1 is denoted by N_i while the total capacity of the (n-1) storage points is denoted by N.

In the following analysis, unit production time, viz., cycle time is taken as a time unit, and transport time between stages is assumed to be negligible or subsumed by the unit production times.

5. 3 DEFINITIONS AND ASSUMPTIONS

The performance of a particular station or stage is described by whether it is :

Operating : in working order and carrying out its function [abbrev. : 1]

Broken down and under repair : Each stage in the line is subject to breakdowns which are random in both occurrence and duration. These breakdowns may be the result of a malfunction, or time required to change or adjust tools, settings, and so forth [abbrev. : O]

Foreced down (type 1) : in working order but unable to operate because it has no workpiece to process. The stage is said to be idle or starved. [abbrev. : I]

It is assumed that if a stage can not transfer it completed workpiece to the next station or has no place to eject it into the buffer store, then the stage holds the workpiece.

Forced down (type 2) : the stage is physically able to produce but it can not transfer its completed workpiece to the next station or into the buffer store. The stage is then said to be blocked. [abbrev. : B]

The role that a buffer plays is to diminish or eliminate the transmission of forced breakdown by means of its storing and replenishing functions. Forced breakdowns transmit either forward or backward. Forward transmission occurs in the case in which a stage cannot operate because it has no workpiece to process, and the forced breakdown of type 1 transmission occurs in the case in which a stage cannot transfer its completed workpiece to the next station or into the buffer store, and the forced breakdown of type 2 transmits simultaneously to the preceding stages. The transmission speed of the former, or conversely speaking, the buffer effect of the former, depends on the number of stocks present in the buffer space. The transmission speed of the latter relates to the spare space in the buffer space. The stages under forced breakdowns of type 1 begin operating successively as repair of the broken down stage is completed. All the stages under forced breakdowns of type 2 begin operating simultaneously as soon as repair of the broken down stage is completed.

The following states describing the line behavior are defined :

Up : The line is considered to be producing whenever the last stage is turning out finished workpieces.

Down : Otherwise it is said that the line is down, either because it has had a breakdown or because some other station in the line has a breakdown and the last station is forced down.

The efficiency and the mean buffer stocks are defined in the following way : The efficiency of the line is the probability that at the steady state the last stage is up. The mean buffer stocks are the expected number of workpieces in the buffer space at the steady state.

On the characteristics of breakdown and repair of the stages, the following fundamental assumptions are made :

(1) On the characteristic of breakdown of the stages : It is assumed that the probability that the stage i breaks down in a cycle given that it was working at the end of the previous cycle is λ_i , which is called breakdown rate. The breakdown rate of a forced down stage is assumed to be zero.

(2) On the characteristic of repair of the stages : It is assumed that the probability that repair of the broken down stage i is completed in a cycle given that it was broken down and under repair at the previous cycle is μ_i , which is called repair rate.

(3) It is assumed that the stage i does not hold its workpiece when it is broken down.

In order to analyze the effect of buffer storage capacity for the production lines satisfying the above assumptions, the problem might be dealt with by considering the number of buffer stocks as a state variable as a most convenient approach since the change of buffer stocks implies the change of the stage state. But, the probability that the number of buffer stocks changes cannot be fixed since the cause that the number of buffer stocks changes does not imply only one definite state change. Therefore, this approach can not, unfortunately, be adopted. In this presentation, the state of the line by (1,0,B,I) of each stage and the number of buffer stocks in each buffer will be defined. Also evaluation of the line by seeking the stationary probabilities at the steady state and calculating the line efficiency and the mean buffer stocks will be explored.

5.4 ANALYSIS

5.4.1 Efficiency of a Single Stage Line

There are two states for a single stage line, viz., an operating state, and a broken down and under repair state. As the stationary probabilities of the two states of the line in the long run are sought (the steady state) the initial state of the line is immaterial.

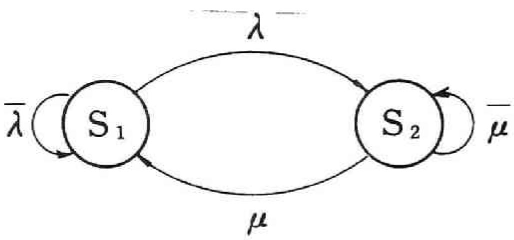


Fig. 5.1. Transition diagram of a single stage line.

Fig. 5.1 shows a schematic transition diagram of the single stage line, where S_1 and S_2 show the states that the stage is operating, broken down and under repair, respectively. Let $\boldsymbol{\pi} = (\pi_1, \pi_2)$ denote the stationary probabilities of the states S_1 and S_2. The matrix of transition probabilities assumes the simple from :

$$T = \begin{pmatrix} \bar{\lambda} & \lambda \\ \mu & \bar{\mu} \end{pmatrix} \quad \begin{matrix} \lambda + \bar{\lambda} = 1 & 0 < \lambda < 1 \\ \mu + \bar{\mu} = 1 & 0 < \mu < 1 \end{matrix} \tag{5.1}$$

where $0 < \lambda < 1$, $0 < \mu < 1$, $\lambda + \bar{\lambda} = 1$ and $\mu + \bar{\mu} = 1$.

Then,

$$T^t = \frac{1}{\lambda+\mu}\begin{pmatrix} \mu & \lambda \\ \mu & \lambda \end{pmatrix} + \frac{(1-\lambda-\mu)^t}{\lambda+\mu}\begin{pmatrix} \lambda & -\mu \\ -\mu & \lambda \end{pmatrix} \tag{5.2}$$

where factors common to all four elements have been taken out as factors to the matrices. Since $|1-\lambda-\mu|<1$, the second matrix tends to zero as $t \to \infty$. Therefore,

$$\lim_{t\to\infty} T^t = \frac{1}{\lambda+\mu}\begin{pmatrix} \mu & \lambda \\ \mu & \lambda \end{pmatrix} \tag{5.3}$$

For an arbitrary initial distribution

$\pi^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)})$, $\pi_1^{(0)} + \pi_2^{(0)} = 1$

$$\lim_{t\to\infty} \boldsymbol{\pi}^{(t)} = \lim_{t\to\infty} \boldsymbol{\pi}^{(0)} T^t = \left(\frac{\mu}{\lambda+\mu}, \frac{\lambda}{\lambda+\mu}\right) \tag{5.4}$$

Therefore, the limiting probabilities of the states do not depend on the initial distribution. Eventually the efficiency of the single line is

$$E = \frac{\mu}{\lambda+\mu} = \frac{1}{1+\rho} \tag{5.5}$$

where $\rho = \lambda/\mu$.

How many cycles are necessary for the line to converge to the above efficiency ? It needs 66 cycles in the case

$\lambda = 1/200$ and $\mu = 1/20$, if the second term in equation (5.2) can be considered negligible when it becomes less than 10^{-3} . This indicates that the line usually converges to the above efficiency quickly.

Now instead of the assumption (2) suppose that it takes a constant time τ for the stage to change from the broken down state to the operating state. A transition diagram can be easily obtained for this case by constructing instead of S_2 the states S_{21} , S_{22} , ... , and $S_{2\tau}$, and the line efficiency can be arrived at in the same manner :

$$E = \frac{1}{1+\lambda\tau} \qquad (5.6)$$

Assuming that the time necessary for the line to change from the broken down state to the operating state follows a normal distribution (μ , σ) results in the same. Consider some long period during which the stage is operating for a total of H cycles. During the period the stage will be broken down for about λH occasions. The total time that the stage is broken down in the period is expected to be $\lambda H\tau$. Hence the efficiency is

$$E = \frac{H}{H+\lambda H\tau} = \frac{1}{1+\lambda\tau} ,$$

which is the same as (5.6) .

This way of thinking is quite convenient and can also apply to the case in which there are n stages without buffers.

5.4.2 Efficiency of a Two Stage Line

(1) $N_1 = 0$.

There are six possible states which are illustrated in Fig. 5.2 for a two stage line without a buffer.

S_1: $\boxed{1 \mid I}$ S_2: $\boxed{1 \mid 1}$

S_3: $\boxed{1 \mid 0}$ S_4: $\boxed{B \mid 0}$

S_5: $\boxed{0 \mid I}$

S_6: $\boxed{0 \mid 0}$

Fig. 5.2. Six possible states for a two stage line without a buffer.

The matrix of transition probabilities assumes the following form :

$$T = \left\{ \begin{array}{cccccc} 0 & \bar{\lambda}_1 & 0 & 0 & \lambda_1 & 0 \\ 0 & \bar{\lambda}_1\bar{\lambda}_2 & 0 & \bar{\lambda}_1\lambda_2 & \lambda_1\bar{\lambda}_2 & \lambda_1\lambda_2 \\ 0 & \bar{\lambda}_1\mu_2 & 0 & \bar{\lambda}_1\bar{\mu}_2 & \lambda_1\mu_2 & \lambda_1\bar{\mu}_2 \\ 0 & \mu_2 & 0 & \bar{\mu}_2 & 0 & 0 \\ \mu_1 & 0 & 0 & 0 & \bar{\mu}_1 & 0 \\ \mu_1\mu_2 & 0 & \mu_1\bar{\mu}_2 & 0 & \bar{\mu}_1\mu_2 & \bar{\mu}_1\bar{\mu}_2 \end{array} \right\} \tag{5.7}$$

Since the Markov process is normal, there exist stationary probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_6)$ of S_1, S_2, ..., and S_6. The solution of the equation :

$$\pi = \pi T \tag{5.8}$$

or

$$\left.\begin{aligned} \pi_j &= \sum_{\nu=1}^{6} \pi_\nu p_{\nu j}, \quad j=1, 2, \cdots, 6 \\ \sum_{j=1}^{6} \pi_j &= 1 \quad , \end{aligned}\right\} \tag{5.9}$$

is

$$\begin{aligned}
\pi_1 &= \frac{\lambda_1 (r-\rho_1 s^2 - qs)}{(1+\rho_1)(r-\rho_1 s^2)+(1-\lambda_1)^2 \rho_2 r} , \\
\pi_2 &= \frac{(1-\lambda_1)(r-\rho_1 s^2)}{(1+\rho_1)(r-\rho_1 s^2)+(1-\lambda_1)^2 \rho_2 r} , \\
\pi_3 &= \frac{\lambda_1 q s}{(1+\rho_1)(r-\rho_1 s^2)+(1-\lambda_1)^2 \rho_2 r} , \\
\pi_4 &= \frac{(1-\lambda_1)^2 \rho_2 r}{(1+\rho_1)(r-\rho_1 s^2)+(1-\lambda_1)^2 \rho_2 r} , \\
\pi_5 &= \frac{\rho_1 (r-\rho_1 s^2)-\lambda_1 q}{(1+\rho_1)(r-\rho_1 s^2)+(1-\lambda_1)^2 \rho_2 r} , \\
\pi_6 &= \frac{\lambda_1 q}{(1+\rho_1)(r-\rho_1 s^2)+(1-\lambda_1)^2 \rho_2 r} ,
\end{aligned} \tag{5.10}$$

where

$$\begin{aligned} p &= 1-(1-\lambda_1)(1-\lambda_2) , \quad q = (1-\lambda_1)\lambda_2 , \\ r &= 1-(1-\mu_1)(1-\mu_2) , \quad s = \mu_1(1-\mu_2) . \end{aligned} \tag{5.11}$$

The efficiency of the line is

$$E = \frac{(1-p)(r-\rho_1 s^2)}{(1-\rho_1)(r-\rho_1 s^2)+(1-\lambda_1)^2\rho_2 r} \tag{5.12}$$

and the mean buffer stocks are

$$M = 0 \quad .$$

Assuming further that no two stages shut down simultaneously results in the reduction of the states from S_1, S_2, ... , and S_6 to S_1 , S_2 , S_4 , and S_5 . The schematic transition diagram is shown in Fig. 5.3 .

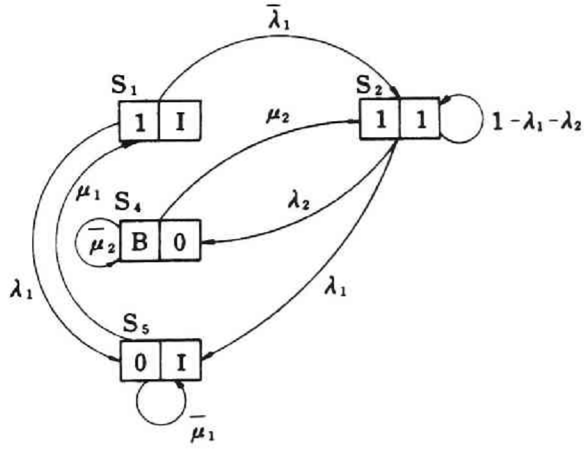


Fig. 5.3. Transition diagram of states S_1, S_2, S_4 and S_5.

The matrix assumes the following form :

$$T = \begin{Bmatrix} 0 & 1-\lambda_1 & 0 & \lambda_1 \\ 0 & 1-\lambda_1-\lambda_2 & \lambda_2 & \lambda_1 \\ 0 & \mu_2 & 1-\mu_2 & 0 \\ \mu_1 & 0 & 0 & 1-\mu_1 \end{Bmatrix} \tag{5.13}$$

Carrying out computation in the same manner results in

$$\left.\begin{aligned} E &= \frac{1-p}{(1+\rho_1)+(1-\lambda_1)\rho_2} \\ M &= 0 \end{aligned}\right\} \tag{5.14}$$

(2) $N_1 \in (0,\infty)$.

First of all, consideration is given to what kinds of states and how many states are possible in a case where the buffer capacity is N_1 . For the previous case in which the buffer capacity is zero, there are six possible states. When there are buffer stocks between the stages, the following stage can continue operating with the buffer stocks, and therefore there are seven basic states which are schematically shown in Fig. 5.4 .

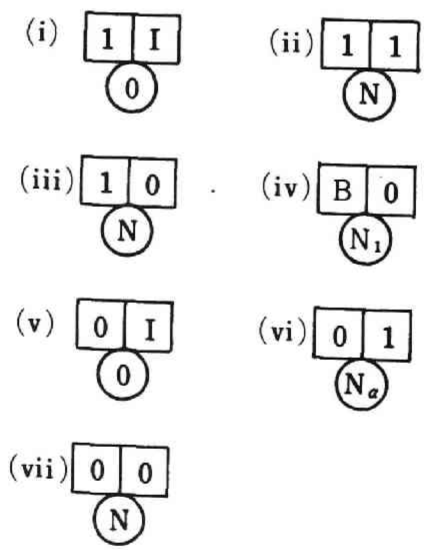


Fig. 5.4. Seven basic states of a two stage line with buffer capacity N_1.

In general if a production line consists of n stages, the number of basic states of the line $B(n)$ is given by the following :

$$B(n) = \left(\frac{3}{4}\sqrt{2}+1\right)(2+\sqrt{2})^{n-1} - \left(\frac{3}{4}\sqrt{2}-1\right)(2-\sqrt{2})^{n-1} \quad . \qquad (5.15)$$

(Proof) Note that three states 1, B, and 0 are possible for the first stage, four states I, 1, B, and 0 for the second ~ (n-1) st stages, and three states I, 1, and 0 for the last stage and that combinations (B ,I), and (B ,1) are prohibited for two consecutive stages. Let $B(n-1)$ denote the total number of the basic states for an (n-1) stage line. Suppose that an n stage line consists of the (n-1) stages and in addition to them the n th stage . There exist at least 3 B (n-1) states since the stages I, 1, and 0 can be added as the n th stage to each of B (n-1) states of the (n-1) stages. Now the (n-1) st stage can be blocked in the n stage line. In this case the n th stage is not operating. The above value 3 B (n-1) does not include this case. The state in which the (n-1) st stage is blocked and the n th stage is not operating can be added to each of B (n-2) states of the (n-2) stages. But the value B (n-2) does not include the case in which the (n-2) nd stage is blocked, the (n-1) st stage is also blocked and the n th stage is not operating. Continuing in the same manner results in the following recurrence formula :

$$B(n) = 3B(n-1) + B(n-2) + \cdots + B(1) + 1 \ . \tag{5.16}$$

The value in the right hand side of the above equation corresponds to the case in which the first ~ (n-1) st stages are blocked and the last stage is not operating. From (5.16),

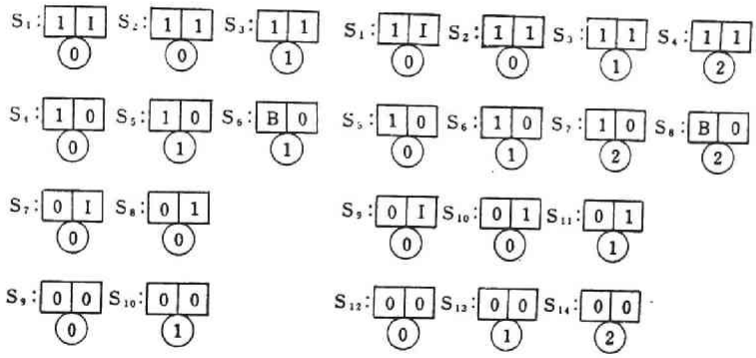
$$B(n) - (2-\sqrt{2})B(n-1) = (2+\sqrt{2})\{B(n-1) - (2-\sqrt{2})B(n-2)\}$$

$$= \cdots = (2+\sqrt{2})^{n-2}(3-2\sqrt{2}) \ , \tag{5.17}$$

where $B(1) = 2$. From (5.17) the above equation (5.15) can be easily obtained.

Taking the number of buffer stocks into consideration in addition to the above basic states results in construction of various possible states. The case of a two stage line with buffer capacity N_1 will be considered next. In Fig.5.4, the cases (i) and (v) in which the second stage is starved occur only when there exists no buffer stock . The case (iv) in which the first stage is blocked happens only when the buffer space is full of buffer stocks. The cases (ii), (iii), and (vii) arise no matter how many buffer stocks there are. The case (vi) also occurs no matter how many buffer stocks there are except for the case that the buffer is full of buffer stocks. Therefore, the total number of possible states is $2(2N_1+3)$ for the two stage line case with buffer capacity

N_1 . It is quite an easy job to obtain the transitional matrix of the case. For example, the possible states and the transitional matrices for the cases of buffer capacity 1 and 2 are shown in Fig. 5.5 and Fig. 5.6, respectively.



(a) Buffer capacity 1 (b) Buffer capacity 2

Fig. 5.5. Possible states of two stage lines with Buffers.

$$\begin{bmatrix}
0 & \bar\lambda_1 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 \\
0 & \bar\lambda_1\bar\lambda_2 & 0 & 0 & \bar\lambda_1\lambda_2 & 0 & \lambda_1\bar\lambda_2 & 0 & \lambda_1\lambda_2 & 0 \\
0 & 0 & \bar\lambda_1\bar\lambda_2 & 0 & 0 & \bar\lambda_1\lambda_2 & 0 & \lambda_1\bar\lambda_2 & 0 & \lambda_1\lambda_2 \\
0 & \bar\lambda_1\mu_2 & 0 & 0 & \bar\lambda_1\bar\mu_2 & 0 & \lambda_1\mu_2 & 0 & \lambda_1\bar\mu_2 & 0 \\
0 & 0 & \bar\lambda_1\mu_2 & 0 & 0 & \bar\lambda_1\bar\mu_2 & 0 & \lambda_1\mu_2 & 0 & \lambda_1\bar\mu_2 \\
0 & 0 & \mu_2 & 0 & 0 & \bar\mu_2 & 0 & 0 & 0 & 0 \\
\mu_1 & 0 & 0 & 0 & 0 & 0 & \bar\mu_1 & 0 & 0 & 0 \\
\mu_1\bar\lambda_2 & 0 & 0 & \mu_1\lambda_2 & 0 & 0 & \bar\mu_1\bar\lambda_2 & 0 & \bar\mu_1\lambda_2 & 0 \\
\mu_1\mu_2 & 0 & 0 & \mu_1\bar\mu_2 & 0 & 0 & \bar\mu_1\mu_2 & 0 & \bar\mu_1\bar\mu_2 & 0 \\
0 & \mu_1\mu_2 & 0 & 0 & \mu_1\bar\mu_2 & 0 & 0 & \bar\mu_1\mu_2 & 0 & \bar\mu_1\bar\mu_2
\end{bmatrix}$$

$$T = \begin{bmatrix}
0 & \bar\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 \\
0 & \bar\lambda_1\bar\lambda_2 & 0 & 0 & 0 & \bar\lambda_1\lambda_2 & 0 & 0 & \lambda_1\bar\lambda_2 & 0 & 0 & \lambda_1\lambda_2 & 0 & 0 \\
0 & 0 & \bar\lambda_1\bar\lambda_2 & 0 & 0 & 0 & \bar\lambda_1\lambda_2 & 0 & 0 & \lambda_1\bar\lambda_2 & 0 & 0 & \lambda_1\lambda_2 & 0 \\
0 & 0 & 0 & \bar\lambda_1\bar\lambda_2 & 0 & 0 & 0 & \bar\lambda_1\lambda_2 & 0 & 0 & \lambda_1\bar\lambda_2 & 0 & 0 & \lambda_1\lambda_2 \\
0 & \bar\lambda_1\mu_2 & 0 & 0 & 0 & \bar\lambda_1\bar\mu_2 & 0 & 0 & \lambda_1\mu_2 & 0 & 0 & \lambda_1\bar\mu_2 & 0 & 0 \\
0 & 0 & \bar\lambda_1\mu_2 & 0 & 0 & 0 & \bar\lambda_1\bar\mu_2 & 0 & 0 & \lambda_1\mu_2 & 0 & 0 & \lambda_1\bar\mu_2 & 0 \\
0 & 0 & 0 & \bar\lambda_1\mu_2 & 0 & 0 & 0 & \bar\lambda_1\bar\mu_2 & 0 & 0 & \lambda_1\mu_2 & 0 & 0 & \lambda_1\bar\mu_2 \\
0 & 0 & 0 & \mu_2 & 0 & 0 & 0 & \bar\mu_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar\mu_1 & 0 & 0 & 0 & 0 & 0 \\
\mu_1\bar\lambda_2 & 0 & 0 & 0 & \mu_1\lambda_2 & 0 & 0 & 0 & \bar\mu_1\bar\lambda_2 & 0 & 0 & \bar\mu_1\lambda_2 & 0 & 0 \\
0 & \mu_1\bar\lambda_2 & 0 & 0 & 0 & \mu_1\lambda_2 & 0 & 0 & 0 & \bar\mu_1\bar\lambda_2 & 0 & 0 & \bar\mu_1\lambda_2 & 0 \\
\mu_1\mu_2 & 0 & 0 & 0 & \mu_1\bar\mu_2 & 0 & 0 & 0 & \bar\mu_1\mu_2 & 0 & 0 & \bar\mu_1\bar\mu_2 & 0 & 0 \\
0 & \mu_1\mu_2 & 0 & 0 & 0 & \mu_1\bar\mu_2 & 0 & 0 & 0 & \bar\mu_1\mu_2 & 0 & 0 & \bar\mu_1\bar\mu_2 & 0 \\
0 & 0 & \mu_1\mu_2 & 0 & 0 & 0 & \mu_1\bar\mu_2 & 0 & 0 & 0 & \bar\mu_1\mu_2 & 0 & 0 & \bar\mu_1\bar\mu_2
\end{bmatrix}$$

(a) Buffer capacity 1 (b) Buffer capacity 2

Fig. 5.6. Transition matrices of two stage lines with buffers.

The objective of displaying these two cases is to show that construction of the possible states for the case of a two stage line with arbitrary buffer capacity N and its transitional matrix are made systematically. After checking the normality of the Markov process and looking for the stationary probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_{4N+6})$, the line efficiency and the mean buffer stocks can be obtained,

$$\mathrm{E} = (1-\lambda_1)\left(\sum_{i=0}^{N} \pi_{i+2} + \sum_{i=1}^{N} \pi_{2N+5+i} \right) \quad , \tag{5.18}$$

$$\mathrm{M} = \sum_{i=1}^{N} i\pi_{i+2} + \sum_{i=1}^{N} i\pi_{N+3+i} + \mathrm{N}\cdot\pi_{2N+4} + \sum_{i=1}^{N-1} i\pi_{2N+6+i} + \sum_{i=1}^{N} i\pi_{3N+6+i} . \tag{5.19}$$

For the case of buffer capacity 1,

$$\begin{aligned} \mathrm{E}(1) &= (1-\lambda_1)(\pi_2+\pi_3+\pi_8) \\ &= \frac{(1-p)\{(r-\rho_1 q^2)u^{(1)} + \mu_2 q v^{(1)} + \alpha_1 qr\}}{(1+\rho_1)\{(r-\rho_1 q^2)u^{(1)} + \mu_2 q v^{(1)} + \alpha_1 qr\} + \alpha_1 t(1-\lambda_1)^2\rho_2 r} , \end{aligned} \tag{5.20}$$

$$\begin{aligned} \mathrm{M}(1) &= 1\cdot\pi_3 + 1\cdot\pi_5 + 1\cdot\pi_6 + 1\cdot\pi_{10} \\ &= \frac{\alpha_1 qr + \{(1-\lambda_1)^2 r + \rho_1\mu_1\mu_2(1-\lambda_1)\}\alpha_1 t\rho_2}{(1+\rho_1)\{(r-\rho_1 q^2)u^{(1)} + \mu_2 q v^{(1)} + \alpha_1 qr\} + \alpha_1 t(1-\lambda_1)^2\rho_2 r} \quad , \end{aligned} \tag{5.21}$$

where the value in the parentheses represent the buffer capacity . For the case of buffer capacity 2,

$$\mathrm{E}(2) = \frac{(1-p)\{(r-\rho_1 s^2)u^{(2)} + \mu_2 q v^{(2)} + \alpha_1(sqr + \mu_1 qt)\}}{(1+\rho_1)\{(r-\rho_1 s^2)u^{(2)} + \mu_2 q v^{(2)} + \alpha_1(\delta qr + \mu_1 qt)\} + \alpha_1 t(1-\lambda_1)^2\rho_2 r} \quad , \tag{5.22}$$

$$\mathrm{M}(2) = \frac{\alpha_1 qr\{\delta + 2 - (1-\lambda_1)v^{(1)}\} - \mu_1 q^2 v^{(1)} + \{2(1-\lambda_1)^2 r + 2\rho_1\mu_1\mu_2(1-\lambda_1) + (1-\lambda_1)\rho_1\mu_1\mu_2\delta\}\alpha_1 t\rho_2}{(1+\rho_1)\{(r-\rho_1 s^2)u^{(2)} + \mu_2 q v^{(2)} + \alpha_1(\delta qr + \mu_1 qt)\} + \alpha_1 t(1-\lambda_1)^2\rho_2 r} , \tag{5.23}$$

where

$$
\left.
\begin{aligned}
\alpha_i &= \frac{1-\lambda_i}{\lambda_i} , \\
\beta_i &= \frac{1-\mu_i}{\mu_i} , \\
t &= \rho_1\{\beta_2+\alpha_1\rho_2\}\mu_1\mu_2 = \lambda_1+\lambda_2-\lambda_1(\lambda_2+\mu_2), \\
u^{(1)} &= 1-\frac{s\cdot t}{r} , \\
v^{(1)} &= 1-\frac{\mu_1\cdot t}{r} , \\
\delta &= \frac{1}{1-\lambda_1}\left\{\frac{\lambda_1 u^{(1)}+q v^{(1)}}{t}\right\} = \frac{1}{1-\lambda_1}\left\{\frac{p}{t}-\frac{\mu_1 t}{r}\right\} , \\
u^{(2)} &= \left\{\delta-\frac{\mu_1\lambda_2}{\beta_2 r}\right\}u^{(1)}+\frac{\mu_1\mu_2\lambda_2}{\beta_2 r} , \\
v^{(2)} &= \left\{\delta-\mu_1\left(1-\frac{\mu_1\lambda_2}{r}\right)\right\}v^{(1)} .
\end{aligned}
\right\}
\tag{5.24}
$$

The efficiency E and mean buffer stocks M of a two stage line can be expressed as functions of buffer capacity in this manner.

Given buffer capacity, breakdown probabilities, and repair probabilities, of a two stage line, looking for the line efficiency and the mean buffer stocks is to solve a set of simultaneous linear equations. In case of buffer capacity N, a set of simultaneous linear equations of (4N+6) variables must be solved. By FACOM 236-60, up to N=36 can be solved from the restriction $4N+6 \leqq 150$. If the line efficiency for buffer capacity more than 36 is wanted, it should be obtained by means of extrapolation. Therefore, the conclusion is that seeking line efficiency and the mean buffer stocks for a two stage line is an easy job.

5.4.3 Efficiency of a Three Stage Line

First, consider how many possible states can be constructed. In general the total number of possible states for an n-stage line with N_i buffer capacity between i th stage and (i+1) st stage (i=1, ... , n-1) is given by the following :

$$P(n) = 2(2N_1+3)(2N_2+3) \ldots (2N_{n-1}+3) \tag{5.25}$$

(Proof by mathematical induction)

Suppose that an n stage line consists of the (n-1) stages and the n th stage. Let $P_I(k)$, $P_1(k)$, $P_B(k)$, and $P_0(k)$ denote the number of possible states up to the k th stage in an n stage line when the k th stage is I, 1, B, and 0, respectively.

Then, following expression is obtained :

$$P(n) = (2N_{n-1}+3)\,P_1(n-1)+(2N_{n-1}+3)\,P_1(n-1) + (2N_{n-1}+2)\,P_0(n-1)+P_B(n-1) \quad , \tag{5.26}$$

where the coefficients of $P_I(n-1)$, $P_1(n-1)$, $P_0(n-1)$, and $P_B(n-1)$ are obtained by considering how the possible number is increased by adding I, 1 and 0 states as the n th stage to the (n-1) stages when the (n-1) st stage is I, 1, and 0, respectively. By mathematical induction, the following may easily be shown :

$$
\left.\begin{aligned}
P_I(n-1) &= (2N_1+3)\cdots(2N_{n-3}+3)\cdot 2 \\
P_1(n-1) &= (2N_1+3)\cdots(2N_{n-3}+3)\cdot(2N_{n-2}+1) \\
P_0(n-1) &= (2N_1+3)\cdots(2N_{n-3}+3)\cdot(2N_{n-2}+3) \\
P_B(n-1) &= (2N_1+3)\cdots(2N_{n-3}+3)\cdot(2N_{n-2}+3)
\end{aligned}\right\} . \qquad (5.27)
$$

By substituting these into (5.26), the equation (5.25) is obtained.

Now, for a three stage line there are $2(2N_1+3)(2N_2+3)$ possible states from the above analysis. It is an easy job to seek the transition matrix for the case. In case of a three stage line with a buffer, the line efficiency and the mean buffer stocks up to buffer capacity of 12 can be obtained by means of FACOM 230-60.

5. 5 CONCLUSIONS

The same argument can be applied to lines with more than 4 stages. Therefore, from an analytical point of view, it is an easy (but tiresome) job to obtain the line efficiency and the mean buffer stocks for a line provided that the number of stages, the probability of breakdown, the probability of repair of each stage, and buffer capacities between stages are known. But from a computational point of view, the task becomes quite horrendous even by means of a modern computer as the number of stages and buffer capacity increase . The observation demands other problem-solving techniques, one of which will be designed and run in Chapter 7 .

CHAPTER 6 OPTIMUM BUFFER INSTALLATION POLICY FOR TWO STAGE LINES

6. 1 INTRODUCTION

Based on the analysis developed in the previous chapter, this chapter will address itself to mainly the following problems about two stage lines.

(1) Should buffer stocks be used ?

(2) What is the effect of given buffer capacity on the line efficiency ?

(3) What is the effect of given buffer capacity on the number of mean buffer stocks ?

(4) What is the effect of variation of breakdown rates with identical repair rates ?

(5) What is the effect of variation of repair rates with identical breakdown rates ?

(6) What is the effect of variation of stage efficiencies?

6. 2 THE LINE EFFICIENCY CURVE AND THE MEAN BUFFER STOCK CURVE AS FUNCTIONS OF BUFFER CAPACITY

In order to get a primary knowledge of the effects of system parameters on the line efficiency and the mean buffer stocks, first balanced systems ($\lambda_1 = \lambda_2 = \lambda$, $\mu_1 = \mu_2 = \mu$, and therefore $\rho_1 = \rho_2 = \rho$) are investigated and then unbalanced systems are considered. For the sake of explanation, the following qualitative adjectives are used to express the values of λ, μ and ρ.

Table 6.1. Qualitative adjectives to express λ, μ, ρ.

	< 1/10000	1/10000 – 1/1000	1/1000 – 1/100	1/100 – 1/10	1/10 – 1.0	> 1.0
λ :	very low	low	medium	high	very high	very high
μ :	very low	very low	low	medium	high	very high
ρ :	very low	very low	low	medium	high	very high

6.2.1 The Line Efficiency for the Case $N_1 = 0$

The line efficiency of a balanced two stage line with no buffer can be calculated by equation (5.12). The results for various breakdown rates and repair rates are shown in Fig. 6.1, and Fig. 6.2. Fig. 6.1 shows the effect of breakdown rates on the line efficiency, while Fig. 6.2, the effect of repair rates on the line efficiency. From the figures, the line efficiency shows a slight falling tendency as the repair rates become high. Particularly when the breakdown rates are high and are greater than about 1/50, this tendency becomes pronounced. But excluding such a case as the line has high or very high breakdown rates, the line efficiency of a balanced two stage line with no buffer may be determined according to the values of ρ as in Table 5.2.

Table 6.2. Efficiency of two stage lines with no buffer

ρ = 1/1000	more than 99%	1/10	more than 80%
1/100	more than 97%	1/5	around 70%
1/50	around 95%	1/2	around 50%
1/20	around 90%	1.0	more than 30%

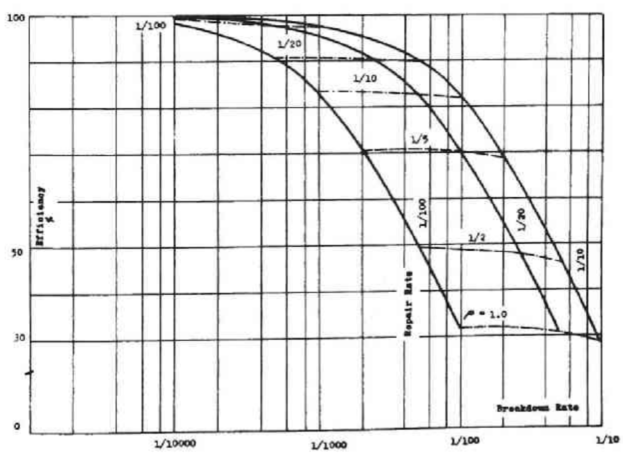


Fig. 6.1. Effect of breakdown rates on the line efficiency of two-stage lines with no buffer.

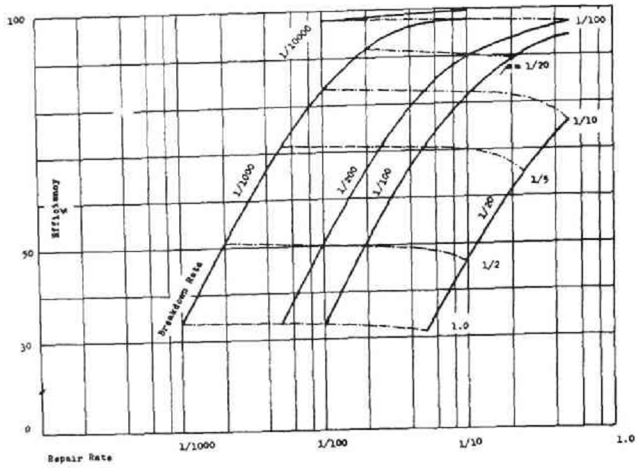


Fig. 6.2. Effect of repair rates on the line efficiency of two-stage lines with no buffer.

The effect of installing a buffer on such a line will be investigated in the following.

6.2.2 The Effect of Installing a Buffer Between the Stages.

The line efficiency and the mean buffer stocks of a two stage line with a buffer can be calculated by equations (5.18) and (5.19). Selecting as the values of ρ, 1/1000, 1/100, 1/10, 1/5 and 1, and choosing as the values of λ and μ the values shown in Table 6.3 and then calculating the line efficiency and the mean buffer stocks by the equations results in Fig. 6.3.

The curves which show the line efficiency, and the mean buffer stocks as functions of buffer capacity are termed the line efficiency curve, and the mean buffer stock curve, respectively.

In the case (1), the efficiency increases by providing buffer capacity of 10, 20, and 30 are 0.04(%), 0.05(%), and 0.06(%), respectively. The effect of improving the line efficiency is a result of the initial buffer capacity of 10, but the efficiency increase itself is quite low.

Table 6.3. System parameters

No.	Conditions			Efficiency	Efficiency with			Buffer Stocks		
	B. R.	R. R.	ρ		10	20	30	10	20	30
(1)	1/10000	1/10	1/1000	99.78	0.04	0.05	0.06	4.8	9.5	14.1
(2)-1	1/10000	1/100	1/100	98.02	0.04	0.08	0.12	5.0	10.0	14.9
-2	1/1000	1/10	1/100	97.84	0.35	0.52	0.62	4.8	9.5	14.1
-3	1/200	1/2	1/100	97.07	0.85	0.95	0.97	3.4	5.2	6.1
(3)-1	1/1000	1/100	1/10	83.18	0.33	0.66	0.99	5.0	10.0	14.0
-2	1/200	1/20	1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
-3	1/50	1/5	1/10	80.25	4.04	5.35	5.95	4.5	8.4	12.1
-4	7/100	7/10	1/10	72.87	5.58	5.75	5.75	2.1	2.4	2.4
(4)-1	1/1000	1/200	1/5	71.30	0.30	0.58	0.86	5.0	10.0	14.0
-2	1/200	1/40	1/5	70.79	1.37	2.46	3.34	4.9	9.9	14.8
-3	1/100	1/20	1/5	70.16	2.34	3.98	5.12	4.8	9.7	14.5
-4	1/50	1/10	1/5	68.91	4.13	6.14	7.31	4.7	9.3	13.8
-5	1/20	1/4	1/5	65.24	6.66	8.41	9.11	4.6	7.4	10.1
(5)-1	1/1000	1/1000	1.0	33.28	0.11	0.22	0.32	5.0	10.0	15.0
-2	1/100	1/100	1.0	32.83	1.04	1.94	2.75	5.0	10.0	14.9
-3	1/20	1/20	1.0	30.87	3.90	6.22	7.76	4.7	9.4	14.0
-4	1/10	1/10	1.0	28.5	5.72	7.07	8.33	4.4	8.3	11.9

The numbers of mean buffer stocks for buffer capacity of 10, 20, and 30 are 4.77 (units), 9.46 (units), and 14.08 (units), respectively, and the mean buffer stock curve is almost linear. The case (2)-1, in which this time the repair rates are 1/10 instead of 1, shows almost the same tendency as the case (1). The case (2)-2 shows slightly better efficiency improvements but the increase is only 0.62 (%) by providing buffer capacity of 30, and also shows the same tendency as the cases (1) and (2)-1. It can be said for such lines that there is little hope of improving the line efficiency and that if efficiency improvement is still required, then a huge amount of buffer capacity must be provided. In the case (2)-3, the efficiency increases by providing buffer capacity of 10, 20, and 30 are 0.85 (%), 0.95 (%), and 0.97(%), respectively. The effect of efficiency improvement is due mainly to the provision of initial buffer capacity of 10. The mean buffer stock curve is markedly different from (1), (2)-1, and (2)-2, and is concave.

The number of mean buffer stocks by providing buffer capacity of 30 is less than 6 (units) which is quite different from about 14 (units) as has been seen in the cases (1), (2)-1, and (2)-2. It can be seen from the figure that the line efficiency curve becomes smooth as the mean buffer stock curve become smooth. This indicates that the repair rates

are so high that it is useless to provide a large amount of buffer capacity to such a line, although the breakdown rates are medium. From this indication it is surmised that it suffices for such a line to provide buffer capacity equal to eight times the mean repair time in order to bring about the effect of installing a buffer in case the breakdown rates are medium. Case (3)-1 is similar to the cases (1), (2)-1, and (2)-2, although the gradient of the line efficiency curve increases slightly higher. In the case (3)-2, the line efficiency curve is concave and shows high efficiency increase. The increases by providing buffer capacity of 10, 20, and 30 are 1.56 (%), 2.61 (%), and 3.34 (%), respectively.
The value of efficiency in case of (3)-2, which has five times higher breakdown rates but five times lesser mean repair time than (3)-1, is greater than that of (3)-1 for buffer capacities more than 6. This means that an unsophisticated line that breaks down often but can be repaired in a short time may achieve higher line efficiency by providing buffer stocks than a sophisticated line that has reduced breakdown rates but required much longer mean repair time. The mean buffer stock curve of (3)-2 is similar to (1), (2)-1, (2)-2, and (3)-1, and is almost linear. The mean buffer stocks for buffer capacity of 30 are around 14. In the case (3)-3, the efficiency increases by providing buffer capacity of 10,

20, and 30 are 4.04 (%), 5.35 (%), and 5.95 (%), respectively.

The effect of providing a larger amount of buffer stock on improvement in line efficiency is marginal. Most of the efficiency is due to the provision of initial buffer capacity of 10. It can be said that it is sufficient to have buffer capacity equal to eight times the mean repair time in order to bring about the effect of providing a buffer when the breakdown rates are high, say, around 1/50. When it comes to the case (3)-4, buffer stocks are hardly saved due to the high breakdown rates of the two stages. The mean buffer stocks for buffer capacity of 30 are around 2 to 3 (units), which is quite low. The efficiency improvement can not be achieved in this case. The stage efficiency of both the stages is 90.9 (%), but linking the two stages causes a remarkable efficiency decrease and gains only 78.6 (%) even by providing buffer storage. The effect of efficiency improvement is due mainly to the provision of initial buffer capacity of 10. From the viewpoint of efficiency, linking stages for such a case should be definitely avoided.

Case (4)-1 shows the same tendency as (1), (2)-1, (2)-2, and (3)-1. In the case (4)-2, the line efficiency curve is near linear and concave, which the mean buffer stock curve is similar to (1), (2)-1, and (3)-1. In the case (4)-3, whose tendency is almost the same as (3)-2, the efficiency increases by providing buffer capacity of 10, 20, and 30 are 2.34 (%), 3.98 (%), and 5.12 (%), respectively. Case (4)-4 shows almost the same tendency as (3)-2 and (4)-3. In the case (4)-5, whose breakdown rates are high and whose repair rates are also high, it shows the similar tendency as (2)-3,(3)-3, and (3)-4. The cases (5)-1 and (5)-2 show the same tendency as (1), (2)-1, (2)-2, (3)-1, and (4)-1 ; the case (5)-3, show the same tendency as (3)-2, (4)-2, (4)-3, and (4)-4, the case (5)-4, show the same tencency as (2)-3, (3)-3, (3)-4 and (4)-5. It suffices to provide buffer capacity equal to ten times the mean repair time in order to produce the effect of installing a buffer in case the breakdown rates are as high as 1/20 or higher than this.

From this investigation, the following two useful results are obtained.

(1) There are three kinds of line efficiency curves and two kinds of mean buffer stock curves in case of balanced two stage lines. In Fig. 6.4 type E-1 represents the case in which the line efficiency curve is approximately linear at first, then tends to turn to be smooth concave and finally converges to a certain value.

In such a line it is almost hopeless to improve the line efficiency by providing a buffer. Type E-2 is the case in which the line efficiency curve shows clear concavity.

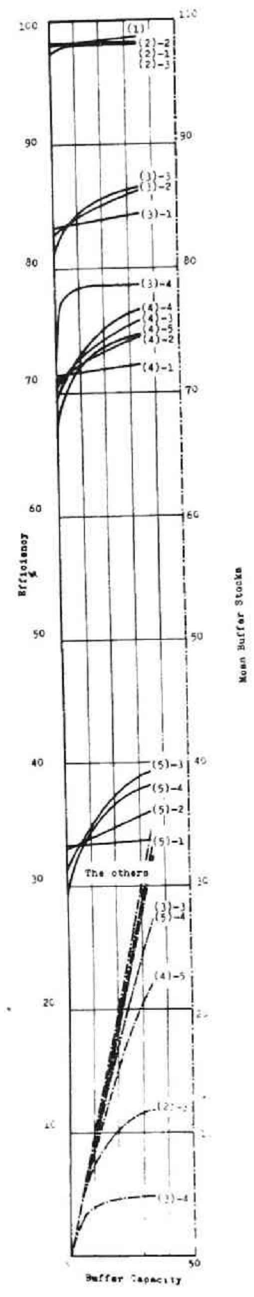


Fig. 6.3. The line efficiency curves and the mean buffer stock curves whose system parameters are shown in Table 6.3.

For such a line the question of buffer capacity arises. Type E-3 shows the case in which at some initial increase of buffer capacity the line efficiency improves markedly, but thereafter the improvements get progressively smaller. The initial buffer capacity which brings about the remarkable efficiency increase depends on the breakdown rates and the repair rates of the two stages, but generally speaking, the initial buffer capacity should equal five times the mean repair time in case the breakdown rates are medium and almost eitht to ten times the mean repair time in case the breakdown rates are high.

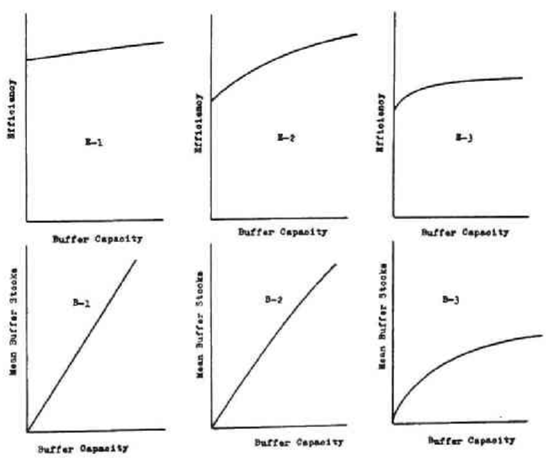


Fig. 6.4. Classification of line efficiency curves and mean buffer stock curves.

The mean buffer stock curves which appear in Fig.6.3 belong to either type B-2 or type B-3 in Fig. 6.4. It will be shown later that type B-1 exists, but for the sake of consistency , type B-1 is explained here with B-2 and B-3. Type B-1 appears when the efficiency of stage 2 is lower than that of stage 1. The mean buffer stocks increase linearly as buffer capacity increases. Type B-2 shows the case in which the mean buffer stock curve is almost linear at first and then turns to be concave and finally tends to converge to a certain value. In the linear portion of the line the mean buffer stocks are about half of the buffer capacity. In case the mean buffer stock curve exceeds this half buffer capacity line, it tends to be of type B-1 and if the mean buffer stock curve goes down the half buffer capacity line, it tends to be of type B-3. Type B-3 represents the case in which at each increase of buffer capacity the rate of increase of the mean buffer stocks gets progressively smaller, and the mean buffer stock curve tends to converge to a certain value. Cases (1), (2)-1, (2)-2, (3)-1, (3)-2, (4)-1, (4)-2, (4)-3, (4)-4, (5)-1, (5)-2, (5)-3 in Fig. 6.3 belong to type B-2, while others, to B-3.

The line efficiency curves and the mean buffer stock curves for buffer capacities up to 35 have been presented before. These calculations revealed that there are three kinds of line efficiency curves and two kinds of mean buffer stock curves subject that both the stages have identical system parameters.

In what follows, buffer capacity of 30 is taken as a measure to investigate the effect of installing a buffer since it suffices to get the line efficiency curve up to buffer capacity 30 in order to judge which type the two stage line concerned belongs to.

(2) Subject to the identical ρ's of two stages, if there is no buffer between the two stages, sophisticated lines having low breakdown rates and long mean repair time have higher line efficiency than unsophisticated lines having higher breakdown rates and shorter mean repair time. But by providing a buffer between the stages, unsophisticated lines may have higher line efficiency than sophisticated lines. However, when it comes to cases such as (3)-4 and (4)-5 in Fig. 6.3 which have high breakdown rates and high repair rates, the line efficiency without buffer is constitutionally low, and therefore expected line efficiency improvement is hardly achieved even providing much buffer capacity.

Now, the line efficiency for buffer capacity of 30 will be investigated. The results are shown in Fig. 6.5, and Fig. 6.6. To compare with the case of no buffer, the cases

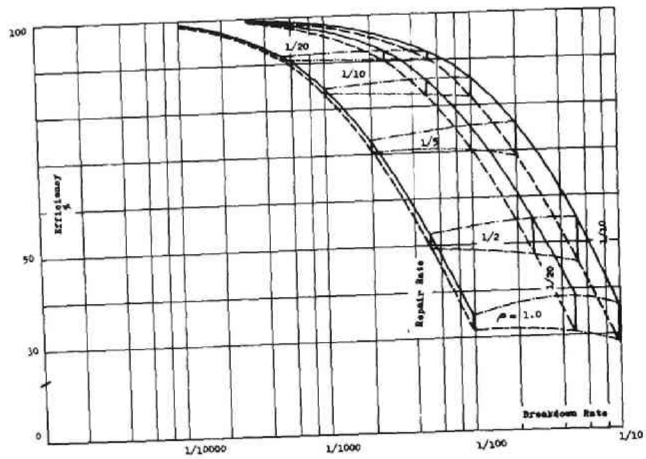


Fig. 6.5. Effects of the breakdown rates on the line efficiency for two-stage lines with buffer.

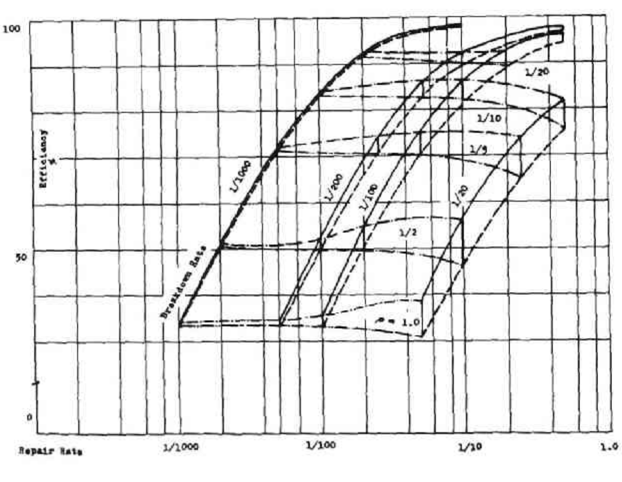


Fig. 6.6. Effects of the repair on the line efficiency for two-stage lines with buffer.

of no buffer are shown by the broken lines, and the cases of having buffer capacity of 30 are shown by the solid lines in the figures. From the figures, it can be said that the positive effects of installing a buffer between stages are relatively certain in case breakdown rates are high and the mean repair time is short.

Fig. 6.7 shows the effects of repair rates on the efficiency increase by buffer capacity of 30. While, Fig. 6.8 shows the effects of breakdown rates on the efficiency increase for buffer capacity of 30. From Fig. 6.7, the effect of providing a buffer is brought about in case breakdown rates are high. From Fig. 6.8, the buffer effect seems to appear mostly when the repair rates are around 1/10, although it depends on the breakdown rates.

6.2.3 The Effects of Variation of Repair Rates with Identical Breakdown Rates

Setting the breakdown rates to be 1/200, and changing the repair rates from 1/2 to 1/200 (Table 6.4.(a), and calculating the line efficiency curves and the mean buffer stock curves results in Fig. 6.9(a). As the value of μ decreases, type E-3 [(1),(2),(3)], then type E-2 [(4),(5)] and finally type E-1 [(6),(7)] characteristically appear. The efficiency increase resulting from providing buffer capacity of 30 gets

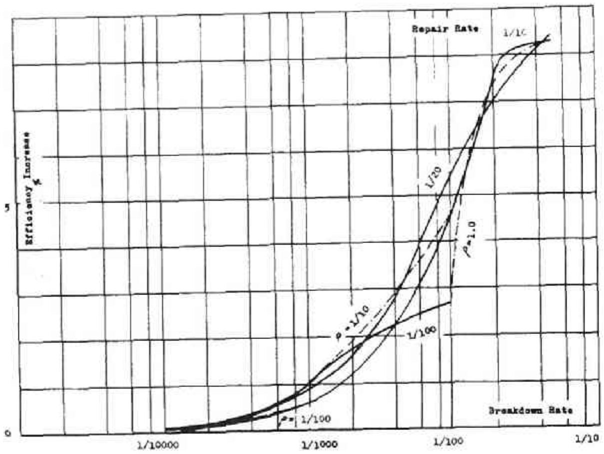


Fig. 6.7. Effects of the breakdown rates on the efficiency increase by providing buffer capacity of 30.

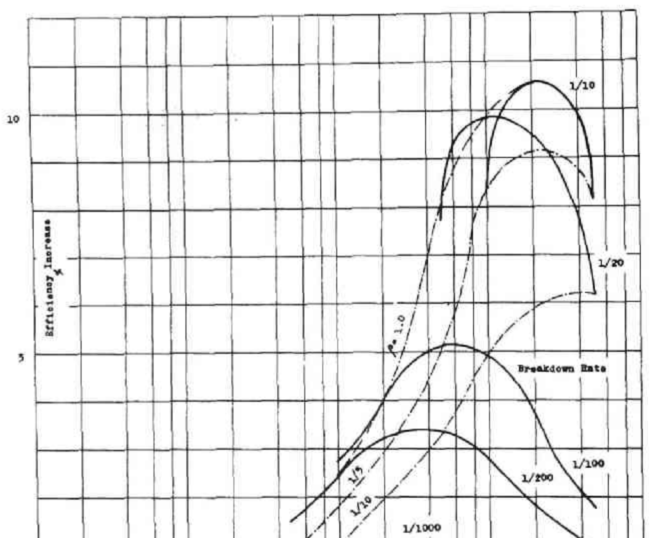


Fig. 6.8. Effects of the repair rates on the efficiency increase by providing buffer capacity of 30.

the maximum value 3.34 (%) when the repair rates are around 1/20. The mean buffer stock curve E-3 shows up, correspondingly with line efficiency curve B-3 ; and E-2 and E-1 appear, correspondingly with B-2.

6.2.4 The Effects of Variation of Breakdown Rates with Identical Repair Rates.

The results by setting the repair rates to be 1/20, and changing the breakdown rates from 1/2000 to 1/20 (Table 6.4.(b)) are shown in Fig. 6.9.(b). As the value λ increases, type E-1 [(1),(2)] appears first and then type E-2 [(3), (4),(5),(6),(7)] shows up characteristically. In this case type E-3 does not turn up. The mean buffer stock curves B-1, and B-2 appear correspondingly with E-1, and E-2, respectively.

Table 6.4. System parameters for variations of repair rates (a) and breakdown rates (b).

	No.	Conditions			Efficiency	Efficiency with			Buffer Stocks		
		B.R.	R.R.	ρ		10	20	30	10	20	30
(a)	(1)	1/200	1/2	1/100	97.07	0.83	0.92	0.94	3.4	5.2	6.0
	(2)		1/4	1/50	95.21	1.18	1.50	1.64	4.3	8.2	11.5
	(3)		1/10	1/20	90.03	1.53	2.29	2.74	4.8	9.3	14.0
	(4)		1/20	1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
	(5)		1/40	1/5	70.79	1.37	2.45	3.33	4.9	9.7	14.7
	(6)		1/100	1/2	49.59	0.89	1.69	2.42	5.0	10.0	14.9
	(7)		1/200	1.0	33.08	0.54	1.04	1.52	5.0	10.0	14.9
(b)	(1)	1/2000	1/20	1/100	97.94	0.20	0.34	0.43	4.9	9.8	14.6
	(2)	1/1000		1/50	95.96	0.40	0.66	0.84	4.9	9.8	14.6
	(3)	1/400		1/20	90.47	0.90	1.49	1.92	4.9	9.8	14.6
	(4)	1/200		1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
	(5)	1/100		1/5	70.16	2.48	4.10	5.52	4.9	9.7	14.5
	(6)	1/40		1/2	47.76	3.59	5.89	7.48	4.9	9.7	14.5
	(7)	1/20		1.0	30.87	5.72	8.07	8.33	4.7	9.4	13.9

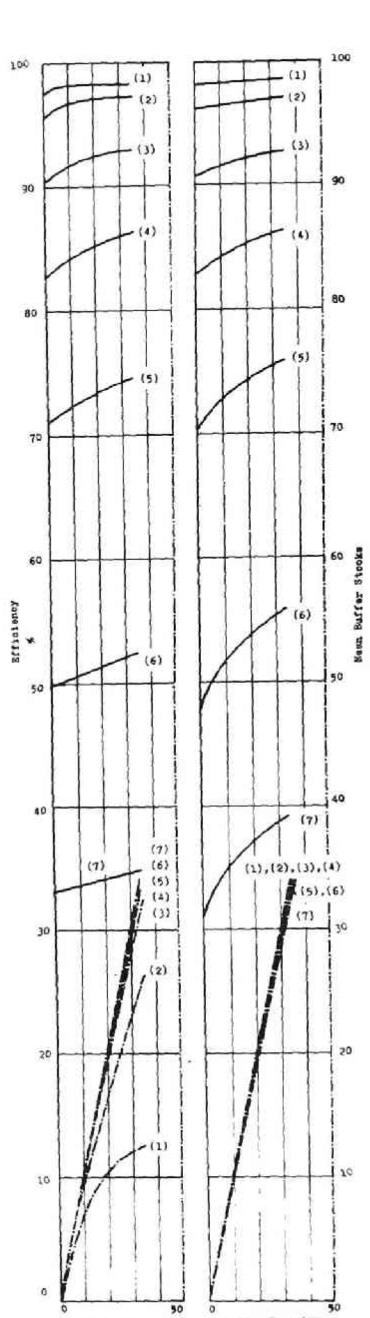


Fig. 6.9. Variations of repair rates (a) and breakdown rates (b).

6.2.5 The System Parameters Which Classify Line Efficiency Curves into Three Types.

It was shown in the previous section that there are three kinds of line efficiency curves. It should be clarified how the system parameters determine these three types. The previous data after classifying them into three types are plotted in Fig. 6.10. In the figure the abscissa represents the breakdown rates, the ordinate, the repair rates. It can be said from the figure that type E-1 (·mark) shows up in case the product of λ and μ is less than $1/10^4$, and that type E-3 (o mark) appears in case the repair rates are less than 1/10 and the breakdown rates are relatively high, and that otherwise type E-2 turns up. When a two stage line has breakdown rates 1/1000 and repair rates 1/10, it becomes type E-1. In this case, as shown in Fig. 6.3, the efficiency increases by providing buffer capacity of 10, 20, and 30 are 0.04 (%), 0.05 (%), and 0.06 (%), respectively, and the effect of improving the line efficiency is mostly due to the provision of initial buffer capacity of 10. From the viewpoint of efficiency increase, the line is of type E-1, but from a relative point of view it has the characteristic of type E-3 also. This is due to low repair rates 1/10 of the line.

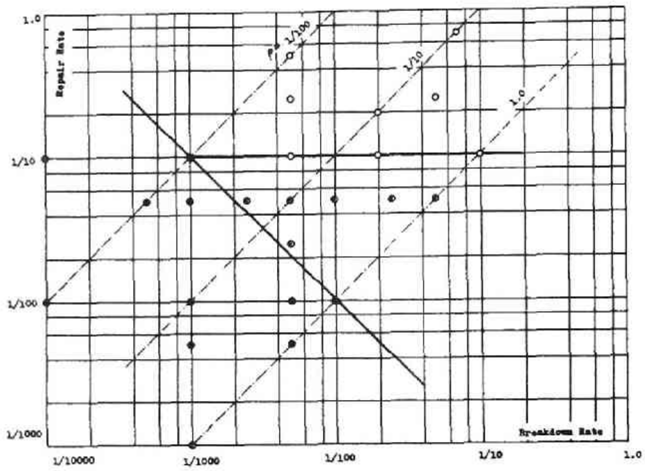


Fig. 6.10. Classification of line efficiency curves by system parameters,

● : type E-1

◐ : type E-2

○ : type E-3 .

6.2.6 The Effect of Variation of Stage Efficiencies.

The primal knowledge on the line efficiency curve and the mean buffer stock curve in the case of a balanced two stage line was obtained in the above. Now the line efficiency curve and the mean buffer stock curve in the case of an unbalanced two stage line will be investigated.

Fig. 6.11(a) shows an example in which both the stages have identical repair rates 1/20 but different breakdown rates 1/400 and 1/200. The detailed system parameters are shown in Table 6.5.(a). For the sake of comparison, the cases where both stages have identical breakdown rates 1/400, and 1/200 are also shown in the figure.

Table 6.5. System parameters for the variations of different breakdown rates.

(a)

	Stags	Conditions			Effi-	Efficiency with			Buffer Stocks		
	No.	B. R.	R. R.	ρ	ciency	10	20	30	10	20	30
(1)	1	1/400	1/20	1/20	90.47	0.90	1.49	1.92	4.9	9.8	14.6
	2										
(2)	1	1/200	1/20	1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
	2										
(3)	1	1/400	1/20	1/20	86.12	1.34	2.07	2.58	6.6	13.4	20.3
	2	1/200	1/20	1/10							
(4)	1	1/200	1/20	1/10	86.12	1.32	1.94	2.53	3.2	6.2	9.0
	2	1/400	1/20	1/10							

(b)

(1)	1	1/200	1/20	1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
	2										
(2)	1	1/100	1/20	1/5	70.36	2.48	4.10	5.52	4.9	9.7	14.5
	2										
(3)	1	1/200	1/20	1/10	75.48	1.92	3.12	3.95	6.6	13.4	20.4
	2	1/100	1/20	1/5							
(4)	1	1/100	1/20	1/5	75.48	2.19	3.37	3.58	3.1	5.1	8.8
	2	1/200	1/20	1/10							

(c)

(1)	1	1/200	1/20	1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
	2										
(2)	1	1/20	1/20	1.0	30.87	5.72	8.07	8.33	4.7	9.4	13.9
	2										
(3)	1	1/200	1/20	1/10	45.18	1.14	1.71	2.00	9.2	18.7	28.3
	2	1/20	1/20	1.0							
(4)	1	1/20	1/20	1.0	45.18	1.06	1.69	2.05	0.7	1.2	1.4
	2	1/200	1/20	1/10							

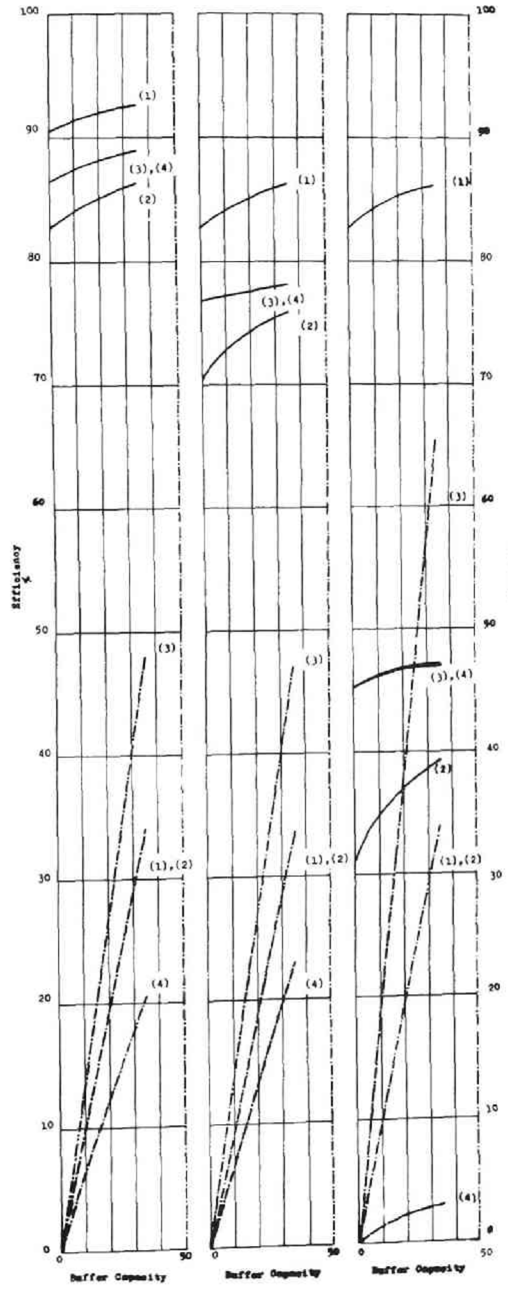


Fig. 6.11. Variations of different breakdown rate.

It can be observed first that from the viewpoint of line efficiency the effects of interchanging stage 1 and stage 2 having different breakdown rates are almost negligible. On the other hand, the mean buffer stock curve in the case (3) in which stage 2 has the higher breakdown rate than stage 1, shows type B-1, the mean buffer stocks by the provision of buffer capacity of 30 being about 20 which is quite high. In the converse case (4), the mean buffer stock curve shows type B-3, the mean buffer stocks by buffer capacity of 30 being around 9 which is quite low. It can be said that in case both stages with identical repair rates have different breakdown rates, the difference between which is not large, the line efficiency takes the intermediate value between the line efficiency gained in the case both the stages have the lower breakdown rates and that gained in the case both the stages have the higher breakdown rates.

Fig. 6.11.(b) shows another example, whose system pareters are shown in Table 6.5.(b). For reference, the cases that both the stages have identical breakdown rates 1/200, and 1/100 are also shown in the figure. From the figure, the difference between the cases (3) and (4) can be neglected. If there is no buffer line efficiency of the cases (3) and (4) takes the intermediate value between the line efficiencies of the cases (1) and (2). If there is a buffer

between the stages, the line efficiency of the case (3) or (4) is affected to a greater extent by the higher breakdown rates. It is quite interesting that both the line efficiency curves of the cases (1) and (2) are of type E-2, but the line efficiency curves of the cases (3) and (4) are of type E-1. According to the classification of line efficiency curves, the breakdown rates and repair rates concerned do not enter into the region that produces E-1 type. It seems to be reasonable that combining two stages which have different breakdown rates results in reducing the effects of installing a buffer. To investigate this matter in detail the case in which two stages with different breakdown rates, the difference between which is large will be examined next. The system parameters are shown in Table 6.5.(c). The results are shown in Fig. 6.11.(c). It can be read from the figure that in case there is a large difference between the two breakdown rates the line efficiency is influenced to a greater extent by the higher breakdown rate, and that the effects of providing a buffer are reduced due to different breakdown rates. The line efficiency curves of the cases (3) and (4) tend to be of type E-3, the gradient of the curve being more smooth than those of the efficiency curves of the cases (1) and (2). Furthermore, it is better for such a line to locate the stage with the lower breakdown rate first.

Finally the following can be concluded for a two stage line with identical repair rates. If the difference of breakdown rates is small, the line efficiency takes the intermediate value between the line efficiency gained in the case where both the stages have the lower breakdown rates and that gained in the case where both the stages have the higher breakdown rates, and the mean buffer stock curve shows type B-1 in the case stage 1 has the lower breakdown rate, and type B-3, otherwise. As the difference increases, the line efficiency is affected to a greater extent by the higher breakdown rate, and the effects of installing a buffer is reduced.

Fig. 12(a),(b), and (c) show three examples in which both the stages have identical breakdown rates 1/200 but different repair rates. The system parameters are shown in Table 6.(a),(b), and (c), respectively. From the figure, the following can be concluded for a two stage line with identical breakdown rates :

If the difference of the repair rates is small, the efficiency takes the intermediate value between the efficiency gained in the case where both the stages have lower repair rates and that gained in the case where both the stages have higher repair rates. As the difference increases, the line efficiency tends to be affected to a greater extent by the lower repair rate, but the effects of installing a buffer are not reduced,

Table 6.6. System parameters for variations of different repair rates.

(a)

	Stage No.	Conditions B. R.	R. R.	ρ	Effi-ciency	Efficiency with 10	20	30	Buffer Stocks 10	20	30
(1)	1 / 2	1/200	1/10	1/20	90.03	1.53	2.29	2.74	4.8	9.3	14.0
(2)	1 / 2	1/200	1/20	1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
(3)	1	1/200	1/10	1/20	86.14	1.55	2.42	2.96	5.5	11.6	18.3
	2	1/200	1/20	1/10							
(4)	1	1/200	1/20	1/10	86.12	1.50	2.32	2.82	4.2	7.6	10.5
	2	1/200	1/10	1/20							

(b)

(1)	1 / 2	1/200	1/20	1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
(2)	1 / 2	1/200	1/40	1/5	70.79	1.37	2.45	3.33	4.9	9.7	14.7
(3)	1	1/200	1/20	1/10	76.21	1.42	2.43	3.18	4.3	8.2	11.7
	2	1/200	1/40	1/5							
(4)	1	1/200	1/40	1/5	76.23	1.45	2.49	3.26	5.5	11.4	17.7
	2	1/200	1/20	1/10							

(c)

(1)	1 / 2	1/200	1/20	1/10	82.55	1.56	2.61	3.34	4.9	9.7	14.5
(2)	1 / 2	1/200	1/200	1.0	33.08	0.54	1.04	1.52	5.0	10.0	14.9
(3)	1	1/200	1/20	1/10	47.27	0.59	1.04	1.38	7.4	15.3	23.6
	2	1/200	1/200	1.0							
(4)	1	1/200	1/200	1.0	47.17	0.57	1.00	1.33	2.5	4.5	6.1
	2	1/200	1/20	1/10							

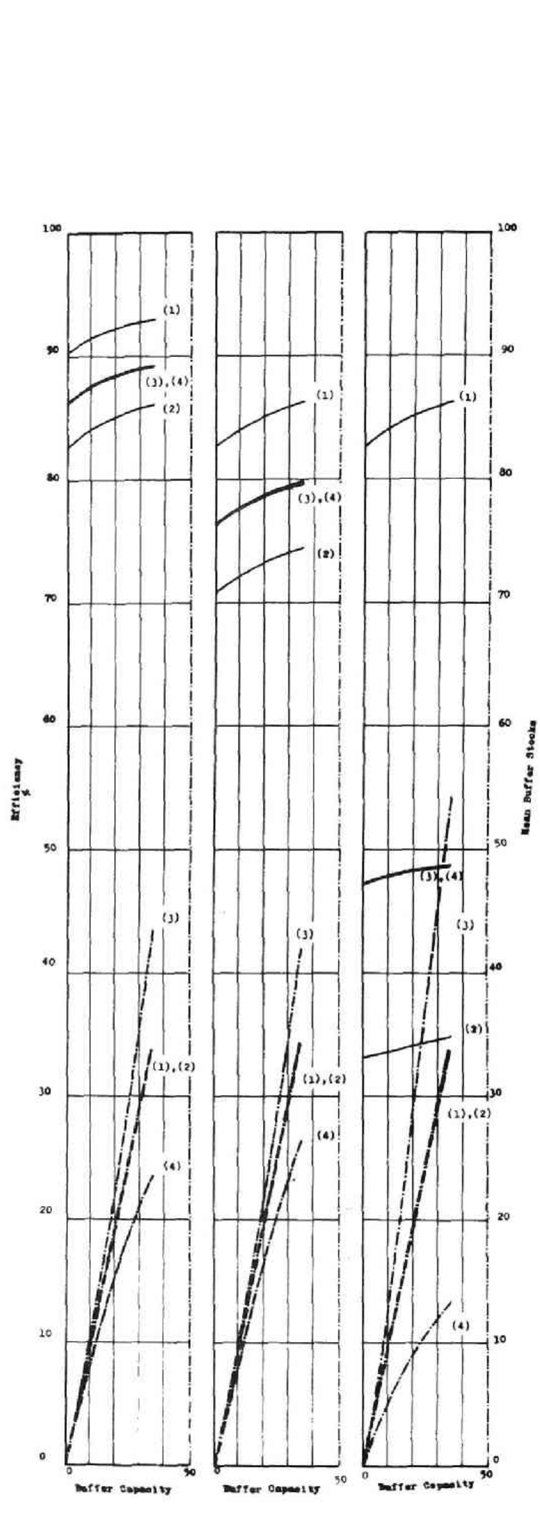


Fig. 6.12. Variations of different repair rates.

as is seen in the case of two stages with different breakdown rates.

From the above, it can be also said that variation of breakdown rates affects the line efficiency more strongly than that of repair rates.

6.2.7 Interchanging the Two Stages Which Have Different Parameters.

As has been seen in Fig. 6.11.(a), (b), and (c), Fig. 6.12. (a),(b), and (c) in Section 6.2.6, from the viewpoint of the line efficiency, the effects of interchanging the two stages which have different parameters are almost negligible. When there is a large difference between two breakdown rates or between two repair rates it is better to locate the stages with the higher stage efficiency first. On the other hand, the mean buffer stock curve is of type B-1 in the case where stage 1 has the higher stage efficiency, and type B -3 in the converse case. The investigation on the line efficiency curve and the mean buffer stock curve for a two stage line is now concluded and an optimal buffer capacity model is to be developed.

6. 3 COST ANALYSIS OF A TWO STAGE LINE

The purpose of this section is to analyze the problem of determining the optimal buffer capacity with respect to the appropriate costs for a two stage line. In order to formulate an optimal buffer capacity model, the following assumptions are introduced.

(1) The revenue from the system is proportional to the line efficiency.

(2) The following two costs are incurred by providing a buffer :

(i) The inventory holding cost, which is proportional to the mean buffer stocks in the buffer storage.

(ii) The storage facility cost, which is the cost to initially install a storage facility plus the cost proportional to buffer capacity.

(3) The profit from the system is given by the following equation :

Profit=Revenue-Inventory holding cost-Storage facility cost.

In order to find the optimal buffer capacity, the line efficiency curve and the mean buffer stock curve as functions of buffer capacity must be known. In the case of two stage lines they are easily obtained as shown in the previous chapter . That is, the line efficiency curve and the mean buffer stock curve up to $N_1 = 36$ are obtained by computer and thereafter they are easily extrapolated, assuming that the line efficiency curve is concave and that the mean buffer stock curve is linear or approximately concave, depending upon the mean buffer stock curve up to $N_1 = 36$. In some cases, however, it is difficult to extrapolate them even with the help of their curves up to $N_1 = 36$. For such cases, they may be obtained by computer simulation to which an approach will be introduced in the next chapter.

Given the line efficiency curve and the mean buffer stock curve, the revenue f(N) and the inventory holding cost h(N) are expressed by the following :

$$f(N) = A \cdot E(N) \text{ [N is buffer capacity]} \qquad (6.1)$$

$$h(N) = B \cdot MW(N) \qquad (6.2)$$

where, A : the fixed revenue in releasing a completed workpiece from the system,

B : the fixed inventory carrying cost per workpiece per time unit,

F(N) : the line efficiency as a function of buffer capacity N,

MW(N) : the mean buffer stocks as a function of buffer capacity N.

The storage facility cost s(N) is expressed by the following :

$$s(N) = \begin{cases} 0 & \text{for } N = 0 \\ C+D\cdot N & \text{for } N > 0 \end{cases} \qquad (6.3)$$

where, C : the fixed cost per unit time of initially installing an inventory storage facility,

D : the fixed cost per workpiece per unit time of maintaining an inventory storage facility.

Therefore, the system profit per unit time $k(N)$ is, then

$$k(N) = f(N) - h(N) - s(N) \qquad (6.4)$$

$$\begin{cases} = A \cdot E(0) - B \cdot MW(0) & \text{for } N=0 \quad (6.5) \\ = A \cdot E(N) - B \cdot MW(N) - C - D\cdot N & \text{for } N>0 \end{cases}$$

The profit curve as a function of buffer capacity can be drawn as shown in Fig. 6.13. The buffer capacity N_0 which gives the maximum of the profit function is the optimal buffer capacity. The problem of installing a buffer comes into consideration only in the case $k(N)-B(0) > C$.

Now assume further that the inventory holding cost is a linear function of buffer capacity, viz., the mean buffer stocks are proportional to buffer capacity, which is estimating the mean buffer stocks higher than the actual values in case the mean buffer stock curve is either type B-2 or type B-3 .

$$k(N) = B' \cdot N \qquad (6.3')$$

Then,

$$\Delta k(N) = \Delta f(N) - B' - N \qquad \text{for } N>0 \quad (6.5')$$

Since function $f(N)$ can be assumed to be concave,

$$\Delta f(N-1) > \Delta f(N) \tag{6.6}$$

Therefore, the buffer capacity which satisfies the following:

$$\Delta k(N_o-1) > 0 > \Delta k(N_o) \tag{6.7}$$

$$\Delta f(N_o) \doteqdot B' + D \tag{6.8}$$

gives the optimal buffer capacity. In other words, the value N_o which satisfies the following on the line efficiency curve

$$E(N_o) \doteqdot \frac{B' + D}{A} \tag{6.9}$$

is the optimal buffer capacity. The equation (6.9) gives a sort of measure up to how much buffer capacity the line efficiency curve should be calculated.

6. 4 CONCLUSIONS

By carrying out investigation on the line efficiency and the mean buffer stocks for various system parameters of two stage lines, the following were obtained :

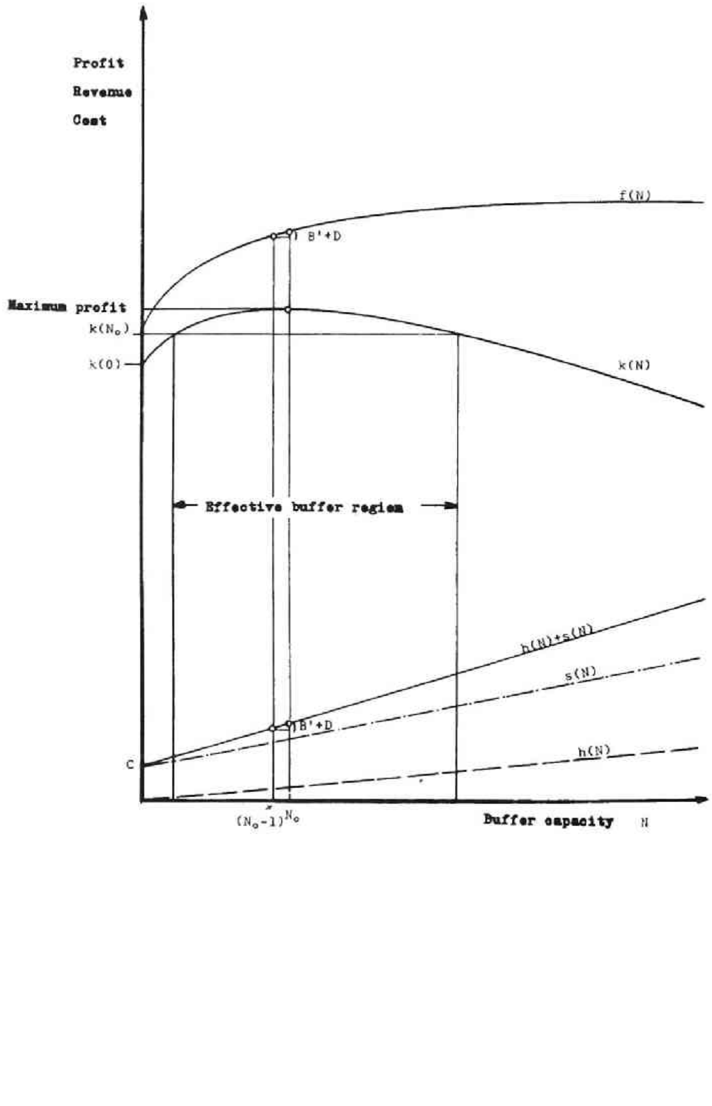


Fig. 6.13. Profit curve as a function of buffer capacity.

(1) In what case should buffer stocks be used ?

There are three kinds of line efficiency curves E-1, E-2, and E-3. In case of two stage lines with identical breakdown rates and identical repair rates, type E-1 appears when the product of λ and μ is less than $1/10^4$ and it is hopeless to improve the line efficiency by providing a buffer in this case ; type E-3 shows up when the repair rates are less than 1/10 and the breakdown rates are relatively high, and in this case the line efficiency improves remarkably at some initial increase of buffer capacity, but thereafter the improvements get progressively much smaller. The initial buffer capacity which brings about the remarkable efficiency increase depends on the breakdown rates and the repair rates of the two stages, but, generally speaking, the initial buffer capacity should be five times the mean repair time in case the breakdown rates are medium, and about eitht to ten times the mean repair time in case the breakdown rates are high ; and otherwise type E-2 turns up, which is clearly concave. In this case the question of how much buffer capacity should be prepared arises and careful cost analysis is required.

(2) As to the mean buffer stocks.

There are three kinds of mean buffer stock curves B-1, B-2, and B-3. Type B-1 represents the case in which the mean buffer stocks increase linearly as buffer capacity increases, Type B-2, the case in which the mean buffer stock curve is almost linear at first and then turns to be smooth concave and finally tends to converge to a certain value. In the linear portion of the line the mean buffer stocks are about half of the buffer capacity ; Type B-3, the case in which at each increase of buffer capacity the increase rate of the mean buffer stocks gets progressively smaller, and the mean buffer stock curve tends to converge to a certain value.

(3) System parameters which demands in-process inventory banks.

Installing buffer storage is most effective when the breakdown rates are high and the mean repair time is short.

(4) The effects of variation of repair rates and breakdown rates.

Variation of breakdown rates affects the line efficiency more strongly than that of repair rates. Therefore, the stages should be designed to have approximately the same breakdown rate.

(5) The effects of variation of stage efficiencies

If the difference of breakdown rates (repair rates) is small, the line efficiency takes the intermediate value between the line efficiency gained in the case where both the stages have the lower breakdown rates (repair rates) and that gained in the case in which both the stages have the higher breakdown rates (repair rates). As the difference increases, the line efficiency is affected to a greater extent by the higher breakdown rate (lower repair rate). The variation of breakdown rates reduces the effects of installing a buffer, while that of repair rates does not.

Cost analysis has revealed the following :

(6) Under the assumption that the mean buffer stock curve is approximately linear, the buffer capacity at which the line efficiency improvement by a unit increase of buffer capacity is equal to (the fixed inventory carrying cost per workpiece per time unit+the fixed cost per workpiece per unit time of maintaining an inventory storage facility) / (the fixed revenue in releasing a completed workpiece from the system) gives the optimum buffer capacity to maximize the system profit.

(7) Since the line efficiency curve is assumed to be a concave function, the system does not demand in-process inventory banks if the line efficiency improvement by providing buffer capacity of 1 does not exceed the above value in (6).

CHAPTER 7 SIMULATIONS FOR ANALYZING THE EFFECTS OF BUFFER STORAGE CAPACITY

7. 1 INTRODUCTION

As has been discussed already, three and more stage model tend to be computationally intractable by the Markov process analysis which was introduced in Chapter 5. The observation establishes the need for other problem-solving approaches, one of which is computer simulation. The obvious limitation of simulation is that it does not provide mathematically proven solutions. However as in the models to be developed in this chapter, the analysis based on the model introduced in Chapter 5 and interpretation of simulation results can provide significant insight into the behavior of multi-stage lines with buffers.

Some initial work involving a simulation approach was performed by Barten[1]. Other research was reported by Freeman[2], Young[3], and Anderson and Moodie[4]. Freeman has described a computer simulation of a three stage line to develop an empirical formula which describes the output rate of particular series of identical finite queue. Anderson and Moodie have developed formulae applicable to their particular simulation studies, but no basis is provided for extrapolating beyond the particular situation studies.

In what follows, the simulation model for production lines having any number of production stages, any size buffer inventory, any breakdown time distribution, and any repair time distribution at any stage, will be developed.

7. 2 SIMULATION MODEL BASED ON THE PREVIOUS MARKOV PROCESS ANALYSIS

To begin with, simulation of the model developed in chapter 5 will be attempted. Simulation for two stage lines is illustrated in the following. Simulations for more than two stage lines are performed in the same manner.

Given the number of stages of a line, the number of basic states of the line is determined. As has been proven in Chapter 5, there are seven basic states of the line. The problem of how the state of a stage is going to change according to the breakdown rate λ and the repair μ , will be tackled first.

For the transition matrix of a stage :

$$T = \begin{array}{c c} & \begin{array}{cc} \underline{S_1} & S_2 \end{array} \\ \begin{array}{c} S_1 \\ S_2 \end{array} & \begin{bmatrix} \bar{\lambda} & \lambda \\ \mu & \bar{\mu} \end{bmatrix} \end{array}$$

where $\lambda+\bar{\lambda}=1$, $\mu+\bar{\mu}=1$, and $0<\lambda,\ \mu<1$,

suppose $\lambda=1/200$ and $\mu=1/20$:

$$T = \begin{bmatrix} 0.995 & 0.005 \\ 0.05 & 0.95 \end{bmatrix}$$

How does the state of the stage change according to the probabilities in each row ? For this case uniform random numbers of three digits for λ and two digits for μ are provided and these are assigned according to the probabilities in the matrix as in the following .

$$T = \begin{bmatrix} 0-994 & 995-999 \\ 0-4 & 5-99 \end{bmatrix}$$

By generating a suitable random number and evaluating it, the state of each stage is transited into a suitable state. Prepairing the transition matrices for the stages and generating proper random numbers (at most two random numbers for a two stage line) according to the probabilities in the matrices, results in transiting the state of the line.

The next problem is how many simulation runs are necessary to get reliable results. In general, to obtain sufficient statistical accuracy for reliable results, a considerable amount of simulation runs are usually necessary. In this representation for each simulation the efficiency of each stage is recorded every 1,000 units. When this value approaches the theoretical efficiency, the simulation runs are assumed to be the necessary simulation runs to evaluate the stage. The maximum value of those is taken to be the necessary simulation runs to evaluate the line efficiency and the mean buffer stocks of the line.

The flow chart of the simulation model for a two stage line is now shown in Fig. 7.1. The following notations and statement are introduced.

o NS: The number of stages in a line.

o MI: The state of stage i.

o NC(I): The buffer capacity in the i th buffer.

o MW(I): The number of buffer stocks in the i th buffer.

o MW(I,J): The number of times that the buffer stocks are J in the i th buffer. J=0,1,...,NC(I).

o R(I): The upper bound to be determined by [1-the breakdown rate λ_i]. For example if λ_1 =1/200, then R(1)=994.

o P(I): The upper bound to be determined by repair rate μ_i . For example if μ_1 =1/20, then P(1)=4.

o N(I): The output of stage i.

o ITM: The timer of simulation runs.

o ITMS: Required simulation runs. For example ITMS=10,000.

o JI: The number of times that the line is in the basic state i. For a two stage line, i=1,2,...,7. 1:(1,I), 2:(1,1), 3:(1,0), 4:(B,0), 5:(0,I), 6:(0,1), 7:(0,0).

o RAND: Random number. The necessary digits of random numbers are varied according to λ and μ . For simplicity all the random numbers are denoted by RAND.

- o STA: ITM $\leqq$ ITMS ?
- o STB: ITM = ITM + 1
- o SMW(I): J = MW(I) and MW(I,J) = MW(I,J)+1, J = 0,...,NC(I).
- o SRR(I): RAND $\leqq$ R(I).
- o SRP(I): RAND $\leqq$ P(I).
- o SNC(I): NC(I) = NC(I)+1 .
- o SJI: JI = JI + 1, For a two stage line I = 1,...,7.
- o SMW(I)G: NW(I) > 0 ?
- o SMW(I)L: MW(I) < NC(I) ?
- o SMW(I)N: MW(I) = MW(I) - 1.
- o SMW(I)P: MW(I) = MW(I) + 1.

In Fig. 7.1, Block 1 reads in parameters of the line for this test: R(1), R(2); P(1), P(2); ITMS; NC(1), once established, are, of course, fixed and constant for the specific test. Block 2 establishes the initial conditions of the line,

ITM = 0, MW(1) = 0; N(1) = N(2) = 0; MW(1,J) = 0,

J = 0,1,..., NC(1); JI = 0, I = 1,..., 7. The initial state of the line is assumed to take the form M1 = 1, M2 = I, and MW(1) = 0. As far as Block 3 is concerned, to avoid redundancy, explanation is limited to the transitions of

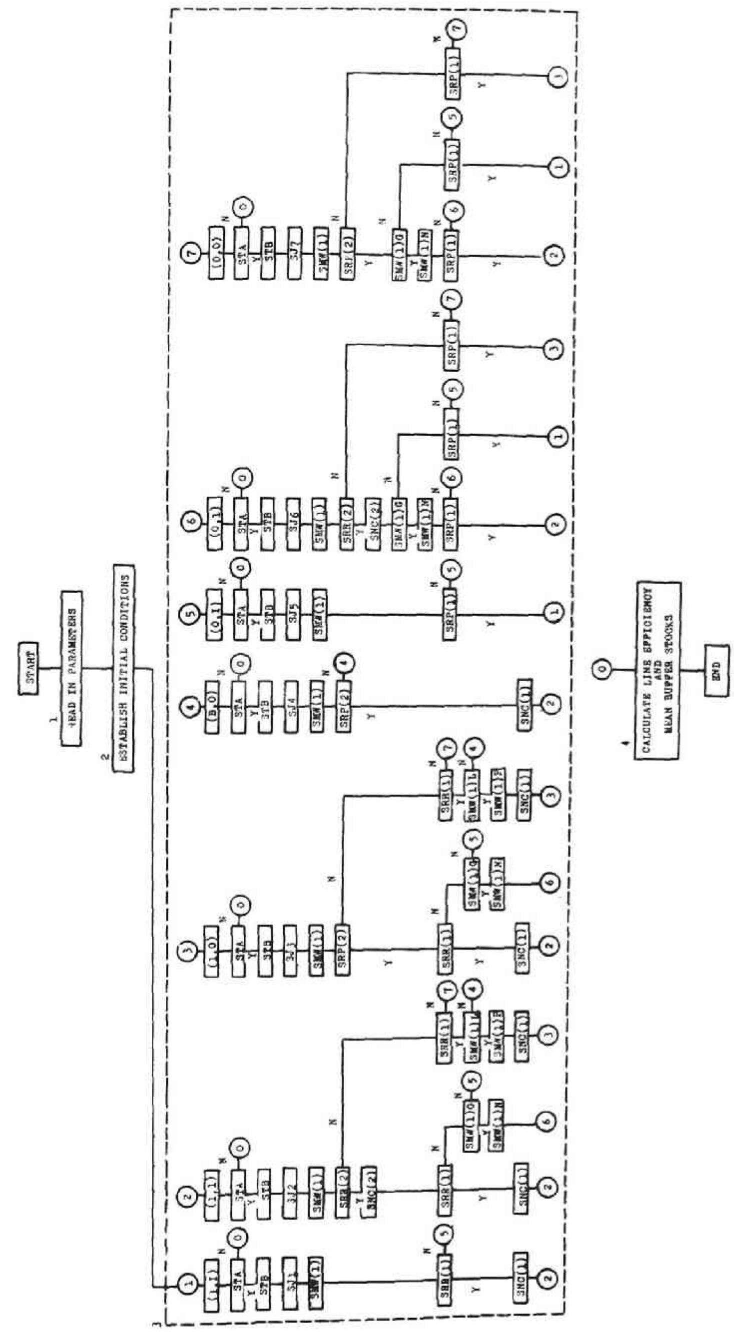


Fig. 7.1. Flow chart of a two stage line.

state 2 of the line, in which the number of buffer stocks is MW(1), according to generated random numbers. First, by the statement STA, it is checked whether the timer exceeds the established simulation runs or not. If the answer is YES (Y), the line efficiency and the mean buffer stocks are calculated in Block 4. If the answer is NO (N), simulation is still continued. Statement STB adds 1 to the timer, SJ2, to the timer of state 2 of the line. Statement SMW(1) performs the same thing about the mean buffer stocks. Statement SRR(2) judges whether or not stage 2 breaks down in the unit time concerned. If the generated random number does not exceed R(2), then stage 2 does not break down and releases one workpiece, which is expressed by SNC(2). SRR(1) performs the same thing about stage 1. If the stage 1 does not break down, the stage releases one workpiece which is expressed by SNC(1), and the state of the line returns to the same state of the line. If the stage breaks down, statement SMW(1)G checks whether or not MW(1) is positive. If MW(1) is positive, then stage 2 can operate in the next cycle. SMW(1)N decreases the number of buffer stocks by 1 and the state of the line changes to state 6. If MW(1) is zero, stage 2 is going to be idle in the next cycle, so the state of the line changes to state 5. Now return to statement SRR(2). If the generated random number exceeds R(2), then stage 2 breaks down in the concerned cycle. SRR(1) checks whether stage 1 breaks down or not. If the generated random number exceeds R(1), stage 1 also breaks down and the state of the line changes to state 7. If the generated random number does not exceed R(1), SMWL judges whether stage 1 can operate without being blocked. If there is not full of buffer stocks in the buffer space, both of MW(1) and the output of stage 1 are increased by 1, and the state of the line shifts to state 3. If the space is full of buffer stocks and there is no room for a new workpiece to be provisioned, stage 1 is blocked and the state of the line goes to state 4. The basic state 2 shifts in this manner to either 2, or 3, or 4, or 5, or 6, or 7 according to random numbers to be generated and the value of mean buffer stocks. The detailed explanation of the shifts of the other basic states is omitted. Block 3 is repeated as many times as required for simulation runs. Block 4 calculates the line efficiency and the mean buffer stocks by the following :

- The line efficiency = $\dfrac{N(2)}{ITMS} \times 100$ (%) ,

- The mean buffer stocks = $\sum_{J=0}^{NC(1)} \dfrac{J \cdot MW(1, J)}{ITMS}$ (unit) .

This is the simulation program for a two stage line.

7.3 AN INVESTIGATION ON STAGE BEHAVIOR

The only problem of extending this simulation model to more than three stages is that it is quite troublesome to prepare for each state of a line the sub-flow chart of its possible changes. For a two stage line, the sub-flow chart must be drawn for each of the seven basic states of the line to examine shifts of basic states. Recall that there are 82 basic states for a four stage line, and 280 basic states for a five stage line. This stipulates the development of a new simulation model easily appliable to lines with more than four stages. To avoid this trouble, it is necessary to know precisely how the behavior of a stage is affected by the preceding stage, the succeeding stage, and the situations in the front buffer and in the back buffer.

The stages of a line can be classified into three groups, the first stage, the second to (n-1)st stages, and the last stage, as shown is the following table.

Table 7.1. Classification of stages.

stage	possible states for the stage
1	0, 1, B
$2 \sim (n-1)$	I, 0, 1, B
n	I, 0, 1

The following is a summary of the stage behavior, according to the classification of stages.

Ⓘ The final stage :

(1) In case the state of the stage is I at time t : (This implies that there is no buffer stock in the (n-1)st buffer).

① If the (n-1)st stage is not producing in the interval between t and (t+1), then the state of the final stage at time (t+1) will remain I.

② If the (n-1)st stage is producing in the interval, then the state of the final stage at time (t+1) will become 1.

(2) In case the state of the stage is 1 at time t :

(i) If the stage does not break down in the interval, then the stage produces one unit.

③ If the (n-1)st stage is not producing in the interval, and if there is no buffer stock in the (n-1)st buffer, then the state of the final stage at time (t+1) will becomes I.

④ If either the (n-1)st stage is producing in the interval, or there are buffer stocks in the (n-1)st buffer, then the state of the final stage at time (t+1) will remain 1.

(ii) Otherwise,

⑤ the state of the final stage at time (t+1) will become 0.

(3) In case the state of the stage is 0 at time t :

(i) If the stage is repaired in the interval ; and

⑥ If the (n-1)st stage is not producing in the interval, and if there is no buffer stock in the (n-1)st buffer, then the state of the final stage at time (t+1) will become I .

⑦ If either the (n-1)st stage is producing in the interval, or there are buffer stocks in the (n-1)st buffer, then the state of the final stage at time (t+1) will become 1.

(ii) Otherwise,

⑧ the state of the final stage at time (t+1) will remain 0.

Ⅱ The i th stage, $i \in [2,3,\ldots,(n-1)]$:

(1) In case the state of the stage is I at time t : (This implies that there is no buffer stock in the (i-1)st buffer).

⑨ If the (i-1)st stage is not producing in the interval, the state of the i th stage at time (t+1) will remain I.

⑩ If the (i-1)st stage is producing in the interval, the state of the i th stage at time (t+1) will become 1.

(2) In case the state of the stage is 1 at time t :

(i) If the stage does not break down in the interval ; and (a) If either the (i+1)st stage is operating or idle in the interval, or there is spare space in the i th buffer; and

⑪ If the (i-1)st stage is not producing in the interval, and if there is no buffer stock in the (i-1)st buffer, then, although the i th stage produces one unit, the state of the i th stage at time (t+1) will become I.

⑫ If either the (i-1)st stage is producing in the interval, or there are buffer stocks in the (i-1)st buffer,

then the i th stage produces one unit and the state of the i th stage at time (t+1) will remain 1.

(b) If the (i+1)st stage is neither operating nor idle in the interval and if there is no spare space in the i th buffer :

⑬ then the state of the i th stage at time (t+1) will be blocked.

(ii) Otherwise,

⑭ the state of the i th stage at time (t+1) will become 0.

(3) In case the state of the stage is B at time t : (This implies that there is no spare space in the i th buffer and that the (i+1)st stage is neither operating nor idle in the previous interval from time (t-1) to t).

(a) If the (i+1)st stage is operating in the interval,

⑮ then the stage releases one unit and the state of the i th stage will become 1.

(b) If the (i+1)st stage is not operating in the interval,

⑯ then the state of the i th stage at time (t+1) will remain B.

(4) In case the state of the stage is 0 at time t :

(i) If the stage is repaired in the interval : and

⑰ If the (i-1)st stage is not producing in the interval, and if there is no buffer stock in the (i-1)st buffer, then the state of the i th stage at time (t+1) will become I.

⑱ If either the (i-1)st stage is producing in the interval, or there are buffer stocks in the (n-1)st buffer, then the state of the i th stage at time (t+1) will become 1.

(ii) Otherwise,

⑲ the state of the i th stage at time (t+1) will remain 0.

Ⅲ The first stage :

(1) In case the state of the stage is 1 at time t :

(i) If the stage does not break down in the interval ; and (a) If either the second stage is operating or idle in the interval, or there is spare space in the first buffer ;

⑳ then the first stage produces one unit and the state of the first stage at time (t+1) will remain 1.

(b) If the second stage is neither operating nor idle in the interval and if there is no spare space in the second buffer :

㉑ then the state of the first stage at line (t+1) will be blocked .

(ii) Otherwise,

㉒ the state of the first stage at time (t+1) will become 0.

(2) In case the state of the stage is B at time t : (This implies that there is no spare space in the first buffer and that the second stage is neither operating nor idle).

(a) the second stage is operating in the interval,

㉓ then the stage releases one unit and the state of the first stage will become 1.

(b) If the second stage is not operating in the interval,

㉔ then the state of the first stage at time (t+1) will remain B.

(3) In case the state of the first stage is 0 :

(i) If the stage is repaired in the interval ;

㉕ then the state of the first stage at time (t+1) will become 1 .

(ii) Otherwise;

㉖ the state of the first stage at time (t+1) will remain 0.

This investigation reveals the important fact that to know whether or not a stage can keep operating, mainly the following two should be examined :

(1) Whether or not can the stage release a completed work-piece ?

(This depends upon the stuations of the succeeding stage and the back buffer).

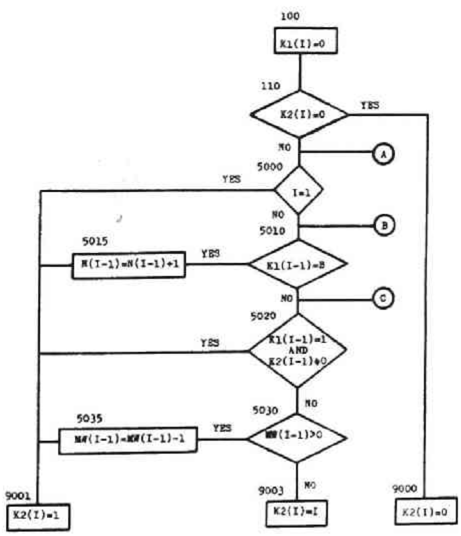
(2) Whether or not can the stage receive a new workpiece ? (This depends upon the situations of the preceding stage and the front buffer).

Note that question (1) does not come up as far as the final stage is concerned since the final stage will deposit the completed workpiece into a storage area which has an infinite capacity, and that question (2) does not arise as far as the first stage is concerned since there is always a supply of workpieces available to the first stage.

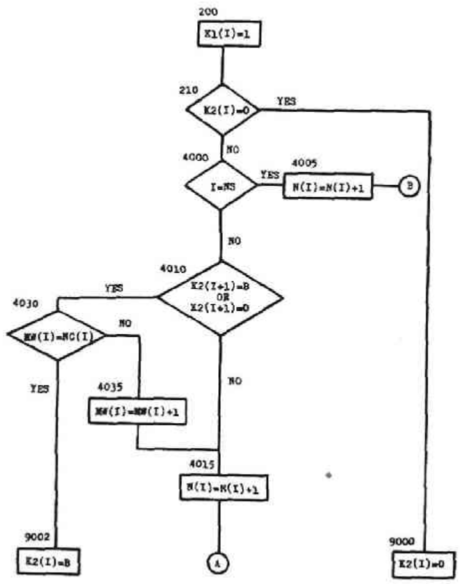
7. 4 THE NEW SIMULATION METHOD AND ITS FLOW CHART

Making good use of the investigation renders it unnecessary to prepare for each state of a line the sub-flow chart of its possible changes. Instead, the behavior of each stage should be examined one by one from the succeeding stages to the preceding stages. Let kl(I), and k2(I) denote the state of the i th stage at the beginning of the t th cycle, and the state of the i th stage at the end of the t th cycle, respectively. Based on the above investigation, the flow charts on the shifts from Kl(I) to k2(I) are shown in Fig. 7.2. In the figure, (a) shows the case that kl(I) is 0 at time t; (b), the case that **kl(I)** is 1 at time t; (c), the case that kl(I) is B at time t ; (d), the case that kl(I) is I, respectively. In the figure (a), Box 5000 and thereafter up to Box 5053 examine whether or not the stage can receive a new workpiece to process. Box 9000 and thereafter represent the state of the stage at the end of the cycle concerned. In the figure (b) Box 4000 and thereafter up to Box 4035 check if the stage can release the completed workpiece, and the Box 5000 and thereafter are exactly the same as Box 5000 and thereafter in the figure (a). In general question (1) should precede question (2). As far as (c) and (d) are concerned, they are assumed to be easily understood.

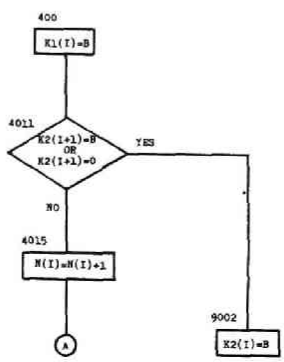
From the viewpoint of simulation, letting kl(I)=k2(I), I = 1,2,..., NS, then means adding 1 to the simulation timer. Then carrying out iterations as many times as required becomes primary concern, which was discussed already is Section 7. 2.



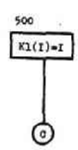
(a)



(b)



(c)



(d)

Fig. 7.2. The shift from K1(I) to K2(I).

Thus the simulation model for production lines having any number of production stages, and any size buffer inventory could be developed. Note that in this simulation model, no assumptions were made on the breakdown time distribution and the repair time distribution of each stage. This means that simulation model can also apply to any breakdown time distribution and any repair time distribution at any stage. For reference a computer simulation program written in FORTRAN for up to 10 stages is given in the appendix. The program is based on the assumptions introduced in Section 5. 2 on the breakdown time distribution and the repair time distribution of each stage.

7. 5 CONCLUSION

It is hoped that the model developed here will help the production line designer make better decisions about the role of buffer stocks in production lines since it is general in the sence that most production lines can be simulated by this model.

```
      DIMENSION MW(9,401),NC(9),N(10),MB(9),K1(10),K2(10),R(10),P(10),WA(9)
      DIMENSION JR(10,2)
 1001 FORMAT(10F8.4/10F8.4)
 1002 FORMAT(20F6.3)
 1003 FORMAT(I2,9I4)
 1004 FORMAT(1HO,10X,I2,6HSTAGES, 5X,15HBUFFER CAPACITY,5X,9I5)
 1007 FORMAT(1HO,20X,2HE=,F6.3,/(1H ,20X,5(3HWA(,I1,2H)=,F8.3,6X)))
      M=100000000
      ITMS=20000
    1 READ(5,1003) NS,(NC(I),I=1,NS-1)
      IF(NS.EQ.0) GO TO 10000
      X=FLOAT(JR(I,1))/FLOAT(M)
      IF(X.GT.P(I)) GO TO 35
      GO TO 45
   35 K2(I)=0
      GO TO 50
   40 IX=JR(1,2)*
      JR(1,2
      X=
                ,NS
             I
        (K1(I)-1) 100,200,300
   00 IF(K2(I).EQ.0) 500,5000
  200 IF(K2(I).EQ.0) 500, 4000
  300 IF(K1(I).EQ.2) 4011,5020
 4000 IF(I.EQ.NS) 4005, 4010
 4005 N(I)=N(I)+1
      GO TO 5010
 4010 IF(K2(I+1).EQ.2.OR.K2(I+1).EQ.0) 4030, 4015
```

```
4011 IF(K2(I+1).EQ.2.OR.K2(I+1).EQ.0) 9002,4015
4015 N(I)=N(I)+1
     GO TO 5000
4030 IF(MB(I).EQ.NC(I)) 9002,,4035
4035 MB(I)=MB(I)+1
     GO TO 4015
5000 IF(I.EQ.1) 9001,5010
5010 IF(K1(I-1).EQ.2) 5015,5020
5015 N(I-1)=N(I-1)+1
     GO TO 9001
```

(

vac

pue

(1

(in

(2

stag

(3)

8. 2

line

buff

thre

B-3;

B-1,

curv

up t

ext

buf

pur

CHAPTER 8 THE BEHAVIOR OF MULTI-STAGE LINES WITH BUFFERS

8. 1 INTRODUCTION

Based on the simulation model developed in the previous chapter, this chapter is to answer mainly the following questions for multi-stage lines :

(1) What are the effects of the number of stages on the relationship between the line efficiency and buffer capacity ?

(2) How should given buffer capacity be allocated among the stages ?

(3) In which order should the stages be placed ?

8. 2 A COMPLEMENT TO THE BEHAVIOR OF A TWO-STAGE LINE

As has been shown in Section 6.2 by calculating the line efficiency curve and the mean buffer stock curve up to buffer capacity N(1)=35 for the two stage line, there are three kinds of line efficiency curves, viz., E-1, E-2, and E-3; and three kinds of mean buffer stock curves, viz., B-1, B-2, and B-3. There, under various system parameters curves E-1, E-2, E-3, B-1, and B-3 seem to have already shown up to buffer capacity of 35 their shapes fully enough to extrapolate thereafter, but curve B-2 has shown up to buffer capacity of 35 only the linear part of it. The purpose of this section is to investigate the mean buffer stock curve thereafter by the first simulation method developed in the previous chapter. Fig. 8.1 shows the case in which both the stages have medium breakdown rates 1/200 and medium repair rates 1/20. It goes without saying that the line efficiency curve shows E-2, the buffer stock curve, B-2.

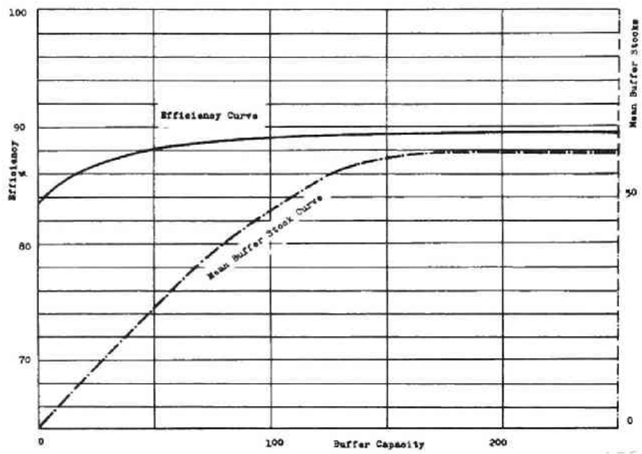


Fig. 8.1. Simulation of a two-stage line, where
breakdown rates : 1/200
repair rates : 1/20

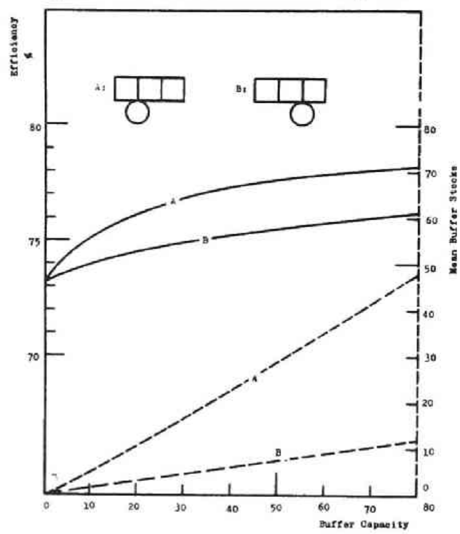


Fig. 8.2. Comparisons of the front buffer and the back buffer.

The buffer stock curve finishes its linear part at about buffer capacity of 120, then turns to becoming smooth concave, and finally tends to converge to a certain value after about buffer capacity of 200.

On the other hand, the line efficiency curve changes accordingly as the mean buffer stock curve changes. From this, it can be said that in case the mean buffer stock curve shows either B-2, or B-3, there is limit in increasing buffer capacity in order to achieve efficiency improvement unless buffer stocks can be adjusted from outside the system.

8. 3 COMPARISONS OF INSTALLING THE FRONT BUFFER AND PROVIDING THE BACK BUFFER AMONG THREE STAGES

This section provides an investigation on the case of a three stage line with one buffer, as an introduction to the studies of multi-stage lines.

Fig. 8.2 shows comparisons of two cases, A, and B; A has the first buffer, and B has the second buffer instead. The system parameters are shown in Table 8.1. In the figure solid lines show the line efficiency curves, and the broken lines, the mean buffer stock curves. In this case, type A yields higher line efficiency than type B. The line efficiency and the mean buffer stocks for the second buffer capacity of $N(2) = 80$ are almost equivalent to those for the first buffer capacity of N(1) = 25 .

Table 8.1. System parameters for comparisons of the front buffer and the back buffer.

Stage No.	B. R.	R. R.	ρ
1	1/200	1/20	1/10
2	1/250	1/25	1/10
3	3/1000	3/100	1/10

The utilization of buffer space :

$$U = [\ MW(I)/NC(I)\ ] \times 100 \quad (\%) \ , \quad I = 1,2 \ . \quad (8.1)$$

of the first buffer is around 50% , while that of the second buffer is quite low. The main purpose of installing a buffer is reducing the number of times of being forced down of type 1 rather than reducing the number of times of being forced down of type 2. Therefore it seems to be reasonable to understand that the low value of U means inefficient use of buffer capacity.

8. 4. THE EFFECTS OF THE NUMBER OF STAGES ON THE RELATIONSHIP BETWEEN THE LINE EFFICIENCY AND BUFFER CAPACITY.

The purpose of this section is to investigate the effect of the way a line is divided into stages. Fig. 8.3 shows the effects of the number of stages on the line efficiency and the mean buffer stocks in case all the stages have identical breakdown rates and identical repair rates and all the buffers have identical buffer capacity. The system parameters are shown in Table 8.2.

Table 8.2 System parameters of multi-stage lines

No.	No. of Stages	B. R.	R. R.
(1)	2, 3, 4, 5	1/1000	
(2)	2, 3, 4, 5	1/200	1/20
(3)	2, 3, 4, 5	1/40	

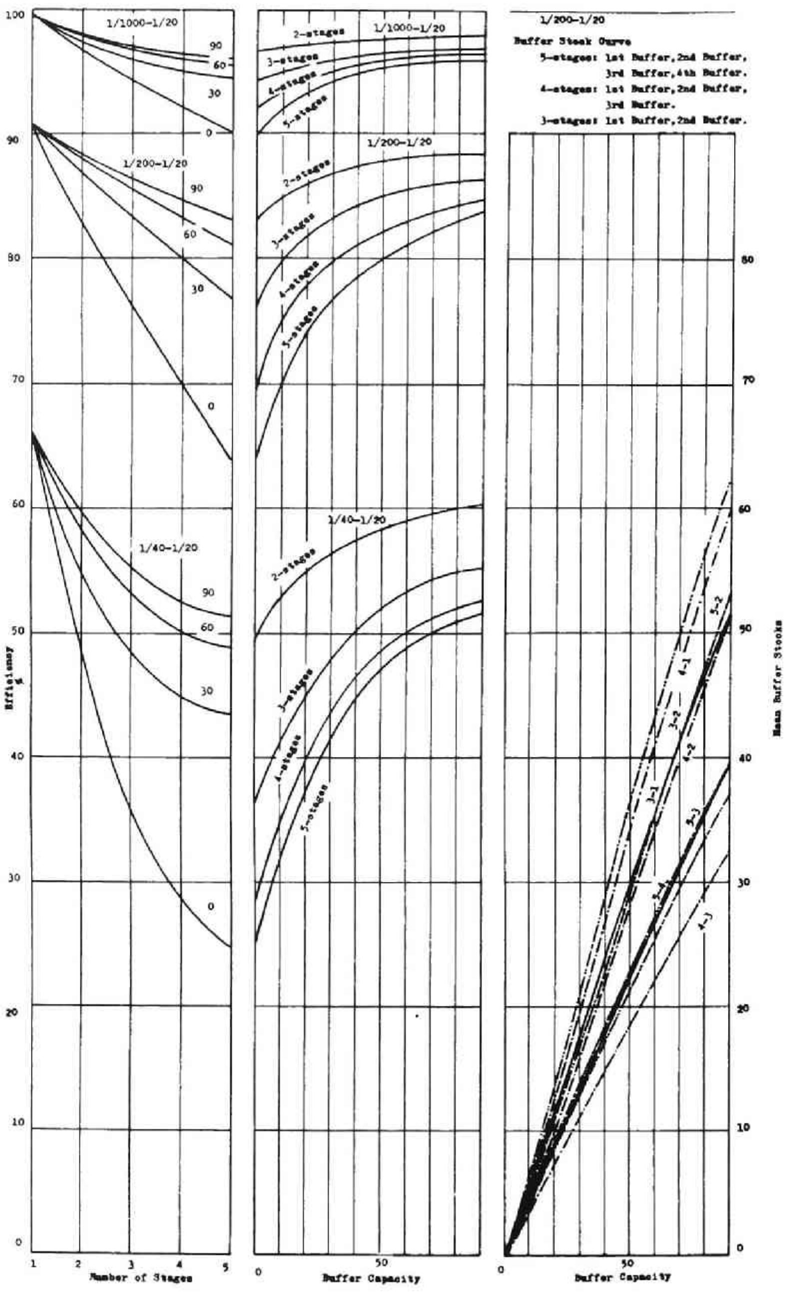


Fig. 8.3. Effects of the number of stages on the line efficiency and the mean buffer stocks.

The repair rates are always set to be 1/20 since this number represents the number which brings about the effect of instal---ling buffer storage, as has been shown in Chapter 6. In the figure (a), the abscissa represents the number of stages, and the ordinate, the line efficiency. The number of stages is of course an integer, but the line efficiency at each number of stages are connected continuously for the sake of easy observation . The cases in which buffer capacity between stages is either 0, or 30, or 60, or 90 are shown in the figure . It can be observed from the figure that in case of no buffer the line efficiency decreases markedly as the number of stages increases. This tendency becomes even more noticeable as the breakdown rates increases. The rate of efficiency decrease per stage decreases as the number of stages decreases. The line efficiency improvement by providing buffer storage can be brought about effectively as the number of stages increases and as the breakdown rates get higher. It can be said from this fact that it is necessary to provide buffer facility in case the breakdown rates of the stages are high. At each increase of buffer capacity, the line efficiency improvement gets progressively smaller, but as the number of stages increases, the buffer capacity which brings about the effect of installing buffer storage increases.

In the figure (b), the abscissa represents buffer capacity, and the ordinate, the line efficiency. Observe the case that breakdown rates are 1/1000. In case of a two stage line, the line efficiency curve of this case shows type E-1, as has been shown in Chapter 6,but changes to type E-2 and E-3 gradually as the number of stages increases. In the cases that breakdown rates are 1/200, and 1/40, both the line efficiency curves for two stage lines, show E-2, but changes to E-3 as the number of stages increases. However buffer capacity at which the line efficiency converges to a certain value increases as the number of stages increases.

Figure (c) indicates, for the example, the mean buffer stock curves of the case that breakdown rates are 1/200. In the figure 5-1, etc., means five stages first buffer,and so forth. In the case breakdown rates are 1/200, the mean buffer stock curve in case of a two stage line is of type B-2 as explained already. Both the mean buffer stock curves 3-1, and 3-2 in case of a three stage line also are of type B-2, but 3-1 tends to be closer to type B-1, and 3-2, to type B-3. In case of a four stage line, 4-1 has a greater gradient than 3-1, but belongs to type B-2 ; 4-2 is of type B-2, but is close to type B-1 ; and 4-3 belongs to type B-2, although it is close to type B-3. In case of a five stage line, 5-1 has still a greater gradient than 4-1, but belongs to type

B-2 ; 5-2, also type B-2, although it is close to type B-1 : 5-3, and 5-4 belong type B-2, although they are close to type B-3. To be more precise, as far as the mean buffer stock curves of the preceding buffers are concerned, they are approximately linear at first, then tend to become concave. On the other hand as far as those of the succeeding buffers are concerned, they are convex at first, then linear and finally tend to become concave. To avoid redundancy, the mean buffer stock curves of the cases in which breakdown rates are 1/2000, and 1/40, are not shown. But the gradient of each mean buffer stock curve in the case that breakdown rates are 1/1000 becomes greater than that of each mean buffer stock curve in the case that breakdown rate are 1/200.
The mean buffer stock curves as to the preceding buffers tend to closer to type B-1. In the case that breakdown rates are 1/40, the gradient of each mean buffer stock curve becomes less than that of each mean buffer stock curve in the case that breakdown rates are 1/200. The mean buffer stock curves as to the succeeding buffers become to closer to type B-3.
It can be concluded from this that when the stages have identical breakdown rates and identical repair rates, it is reasonable to allocate more buffer capacity to the preceding buffers.

8.5 ALLOCATION OF GIVEN BUFFER CAPACITY AMONG THE STAGES

The purpose of this section is to consider how to allocate given buffer capacity among the stages and in which order to place the stages. To answer these questions, three stage lines with two buffers will be investigated since they provide the basis of buffer allocation problem.

In order to investigate the effects of installing buffer storage on the line efficiency and the mean buffer stocks, it is better to change breakdown rates rather than repair rates. Prepair rate 1/20 is the suitable number to check the effects of buffer. From this viewpoint, system parameters shown in Table 8.3 are chosen.

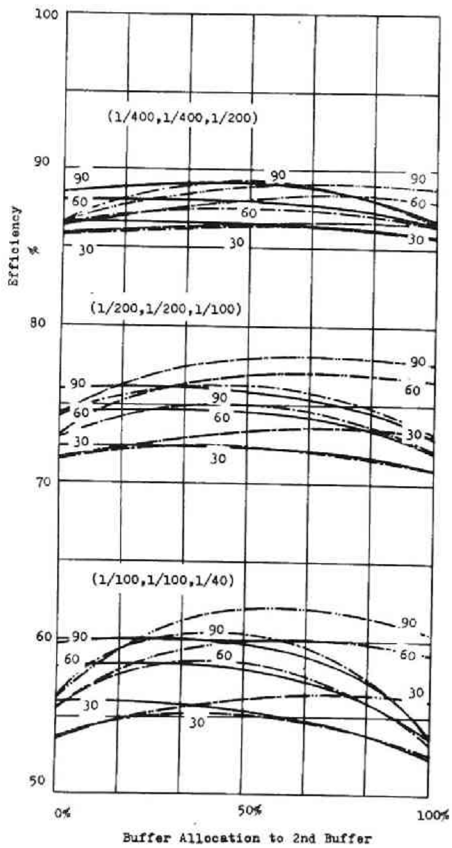
Table 8.3. System parameters for three stage lines

No.	No.of Stages	B. R. Permutation of	R. R.
(1)		(1/400,1/400,1/200)	
(2)	3	(1/200,1/200,1/100)	1/20
(3)		(1/100,1/100,1/40)	

Fig. 8.4 shows the effect of allocating given buffer capacity on the line efficiency. The abscissa represents buffer allocation to the second buffer (%), and the ordinate, the line efficiency. The cases in which the total buffer capacity is either 0, or 30, or 60, or 90 are shown in the figure. The meaning of (1/400, 1/400, 1/200), etc., is that

the repair rates of the stages are identical (1/20) and that one stage has breakdown rate 1/200 and the other two stages have breakdown rates 1/400, and so forth. The solid lines express the cases in which the stage with the highest breakdown rate is placed first; The dot-dash-lines, the cases in which it is placed in the middle ; The dots-dash-lines, the cases in which it is placed last.

First, the solid lines are investigated. In the case (1/400,1/400,1/200), in order to obtain the maximum line efficiency, all of the capacity should be allocated to the first buffer given that the total buffer capacity is 30 ; More than three fourths of the capacity should be allocated to the first buffer subject that the total buffer capacity is 60 ; Less than three fourths of the capacity should be allocated to the first buffer provided that the total buffer capacity is 90. This indicates that allocation ratio of the total buffer capacity to the second buffer varies, depending on the total buffer capacity. When the first buffer capacity is small, the efficiency improvement produced by each increase of buffer capacity in the first buffer is greater than that brought about by each increase of buffer capacity in the second buffer. As the first buffer capacity increases , the efficiency increase gets progressively smaller. When it becomes less than or equal to that produced by each increase of buffer capacity in the second buffer, allocation of buffer capacity to second buffer should be started. This allocation ratio tends to converge to a certain value as the efficiency improvement at each increase of buffer capacity



ig. 8.4. Effects of allocating given buffer capacity on the line efficiency.

becomes negligible. Generally speaking, the problem how much buffer capacity with allocation ratio less than the convergence value should be provided in order to achieve the expected line efficiency, comes into question.

In the case of the two stage line with identical repair rates 1/20 , the breakdown rate at the first stage and the breakdown rate 1/400 at the second stage, the efficiency increases by providing buffer capacity of 10, 20, and 30 are 1.32 (%), 1.94 (%), and 2.52 (%), respectively, as has been shown in Table 6.5.(a).

On the other hand, in the case of the two stage line with identical repair rates 1/20 and identical breakdown rates 1/400, the efficiency increases for buffer capacity of 10, 20, and 30 are 0.90 (%), 1.49 (%), and 1.92 (%), respectively, from Table 6.4.(b).

The combination of a two stage line (1/400, 1/400) and a two stage line (1/400, 1/200) is not equivalent to a three stage line (1/400, 1/400, 1/200), because the assumptions, such as there is always a supply of workpieces available to the first stage and the second stage will deposit the completed workpiece into a storage area which has an infinite capacity, may not hold good.

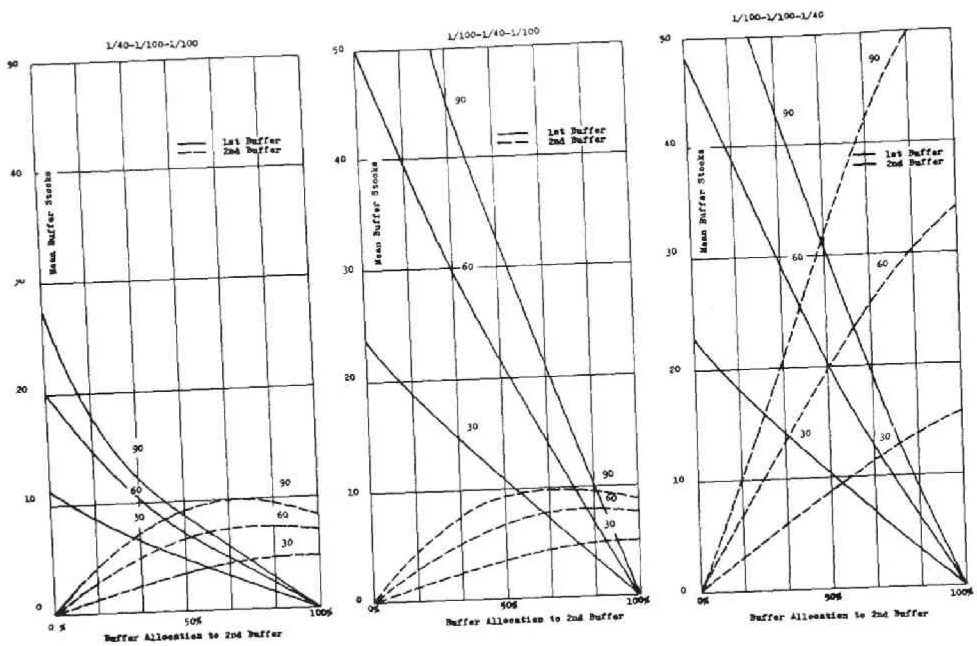


Fig. 8.5. Mean buffer stocks in the front buffer and in the back buffer for the three cases.

But, it may be possible to make use of the effects of buffer capacity on the line efficiency for two stage lines as a sort of criterion to consider buffer capacity allocation to the two buffers in the three stage line. Roughly estimating from this argument in order to obtain the maximum line efficiency, two thirds of the buffer capacity should be allocated to the first buffer provided that total buffer capacity is 30. However, as has been shown above, all of the buffer capacity must be allocated to the first buffer under the same conditions. This indicates that more buffer capacity should be allocated to the preceding buffer. Observing the solid lines of the cases (1/200, 1/200, 1/100) and (1/100, 1/100, 1/40) results in the same conclusion.

Now, the cases in which the stage with the highest breakdown rate comes in the middle, drawn by dot-dash-lines in the figure, are to be investigated. In order to obtain the maximum line efficiency, about seven twelves of the total buffer capacity should be allocated to the first buffer, in any case of (1/400, 1/400, 1/200), (1/200, 1/200, 1/100), and (1/100, 1/100, 1/40). This indicates that it is necessary to provide buffers in front of the bad stage and at the back of it, but it is difficult to read from the figure which buffer, front or back, of the bad stage, more buffer capacity should be allocated to.

Generally speaking, more buffer capacity should be allocated to preceding stages, and therefore it is surmised that more buffer capacity should be given to the front buffer of the stage. Finally, the cases in which the bad stage is placed last, drawn by dots-dash-lines in the figure, are to be examined. Compared with the previous cases, each case of the (1/400, 1/400, 1/200), (1/200, 1/200, 1/100), and (1/100, 1/100, 1/40) achieves the higher line efficiency. In case of two stage lines, from the viewpoint of the line efficiency the effects of interchanging the two stages having different parameters are almost negligible. But, as the number of stages increases, the placement of the stages comes to influence the line efficiency. Particularly in the case (1/100, 1/100, 1/40), the placement effects are pronounced. In order to investigate this matter in detail, the mean buffer stocks in the first, and the second buffer are calculated. Fig. 8.5.(a),(b), and (c) show the mean buffer stocks of the three cases. Figure (a) shows the case in which the bad stage comes first ; in (b), the bad stage comes in the middle; in (c), the bad stage is placed last. Case (c) indicates that more buffer stocks in the first, and the second buffers are saved than either of case (a) and case (b). This shows that buffer installation in the case (c) produces better effects. It can be concluded that unless technical considerations primarily determine the division of the line, the bad stage should follow other better stages.

8. 6 THE EFFECTS OF VARIATION OF BREAKDOWN RATES IN MULTI-STAGE LINES

Finally, the effects of variation of breakdown rate are to be investigated. For the purpose, the effects of the total buffer capacity on the line efficiency are examined, by changing the ratio of the breakdown rates. Fig. 8.6 shows the comparisons of the two cases (1/200, 1/200, 1/40) and (1/100, 1/100, 1/40). The placement of the stage for the two cases is in such an order that the bad stage comes last. In the case (1/200, 1/200, 1/40), all of the buffer capacity should be allocated to the second buffer to achieve the highest line efficiency for any case of buffer capacity of 30, 60, and 90. The line efficiency improvement of the case is less, compared with the case (1/100, 1/100, 1/40) or the cases (1/400, 1/400, 1/200) and (1/200, 1/200, 1/100) in Fig. 8.3. It can be said that different breakdown rates result in reducing the effects of installing buffers, as has been pointed out already in the case of two stage lines.

The investigations on the effects of the number of stages, on buffer capacity allocation among the stages, and on the placement of the stages are now concluded and an optimal buffer capacity model for multi-stage lines will be developed.

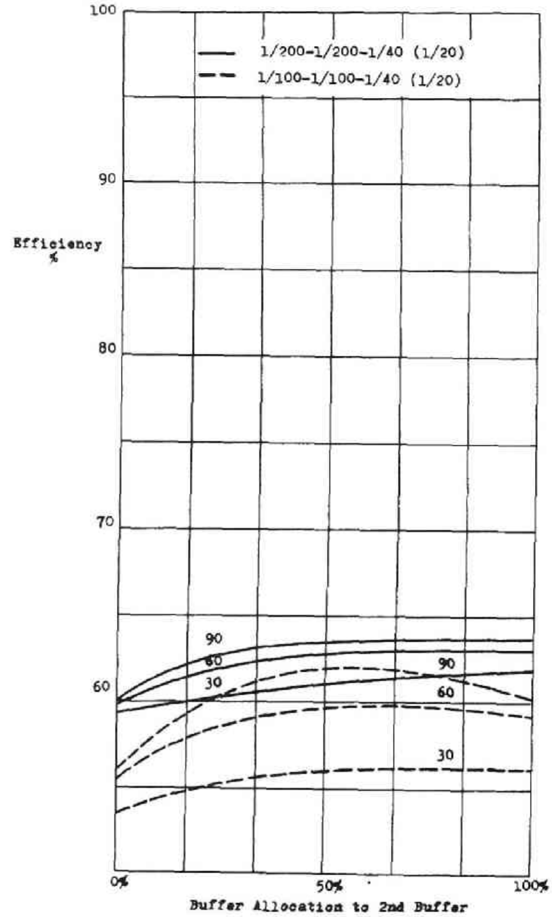


Fig. 8.6. Variation of different breakdown rates.

8. 7 COST ANALYSIS OF THREE-STAGE LINES

The main differences of a three stage line case from a two stage line case in formulating a cost analysis model are that the line efficiency curve and the mean buffer stock curve are obtained for each of the following :

A buffer is provided between stage 1 and stage 2,

A buffer is given between stage 2 and stage 3,

Two buffers are installed among three stages,

and that the storage facility cost s(N) is expressed instead of (6.3) by the following :

$$s(N) = \begin{cases} 0 & \text{for } N = 0 \\ C \cdot M + D \cdot N & \text{for } N > 0 \end{cases} \tag{8.2}$$

where, N : the total buffer capacity,

M : the number of buffers.

In case of a three stage line, it can be said without loss of generality that either the first buffer or the second buffer produces better line efficiency than the other. One example has been shown already in Fig. 8.2, where A brings about greater line efficiency than B, and has more mean buffer stocks at any buffer capacity. As far as inventory holding cost is concerned, it costs more as the mean buffer stocks increase. But, generally speaking, so long as the revenue increases from the line efficiency improvement by each buffer capacity increase is high, inventory holding cost may be relatively small. Therefore, from an economic point of view, installing the buffer which produces less line efficiency such as B in Fig. 8.2 , should be rejected. So, the problem is to discuss how many buffers, either one buffer or two buffers, should be used. When the revenue increase from the efficiency improvement by installing an extra buffer in addition to one buffer exceeds the fixed cost per unit time of initially installing an inventory storage facility C the problem of installing two buffers comes into question. Fig. 8.7 shows the case in which two buffers bring about greater profit. Note that the line efficiency curve obtained by optimum allocation of buffer capacity to two buffers may sometimes cover the line efficiency achieved by installing only one buffer. For example, in the case (1/200, 1/200, 1/40) in Fig. 8.6, the maximum line efficiency is achieved by allocating all of the buffer capacity to the second buffer. Even in such a case, the double times of the fixed cost C are reduced from the revenue gained by installing two buffers. However, this worry is unnecessary, because the revenue line, obtained by installing only one buffer, expressed as a function of buffer capacity includes such a case. Cost analysis for a three stage line is performed in this way, slightly more troublesome, compared with that for a two stage line, but is almost the same as that. Cost analysis for more than three stages is performed in the same manner.

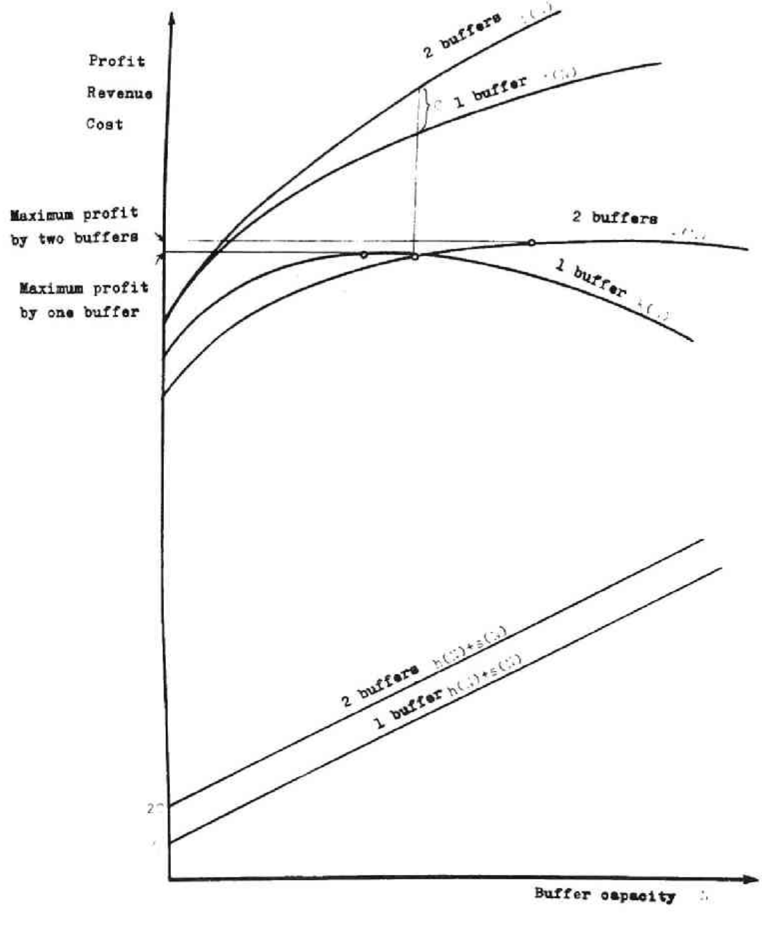


Fig. 8.7. Cost analysis for a three stage line.

8. 8 CONCLUSIONS

By carrying out investigation on the line efficiency and the mean buffer stocks for various system parameters of multi-stage lines by computer simulation, the following were obtained :

(1) How should buffer stocks be used ?

If there is no buffer, the line efficiency decreases markedly as the number of stages increases. This tendency becomes even more noticeable as the breakdown rates increase. The efficiency decrease rate per stage decreases as the number of stages decreases. The line efficiency improvement by providing buffer storage can be brought about effectively as the number of stages increases and as the breakdown rates get higher. As far as the line efficiency curve is concerned, as the number of stages increases, the line tends to be of type E-2 rather than type E-1, or type E-3 rather than type E-2. Therefore, as the number of stages increases, it becomes necessary to provide buffer facility if the breakdown rates of the stages are high. At each increase of buffer capacity, the line efficiency improvement gets progressively smaller, but as the number of stages increases, the buffer capacity which brings about the effects of installing buffer storage will increase .

(2) How should given buffer capacity be allocated among the stages ?

Allocation ratio of the total buffer capacity to the buffers varies, depending on the total buffer capacity. This allocation ratio tends to converge to a certain value as the total buffer capacity increases.

When the stages have identical breakdown rates and identical repair rates, it is reasonable to allocate more buffer capacity to the preceding buffers.

(3) In which order should the stages be placed ?

Unless technical considerations primarily determine the division of the line, the bad stage should follow other better stages.

CONCLUSIONS

In the study of sequencing and in-process inventory control of production lines, certain ways of finding sequences which are in some sense optimal and the insights into the role of inventory banks of production lines have been discussed. The aims of this research are broader than just trying to find algorithms of finding preferable sequences and to give better guidance on the role of buffer stocks of production lines. As always, many important and interesting questions remain as yet unanswered.

In Part I, the sequencing problems of production lines have been discussed. Management is responsible not only for constructing a feasible and realistic sequence but also for analyzing and evaluating all possible alternatives in order to determine which one to adopt. Of the many sequencing problems, two basic sequences, linear sequences and compound sequences, were considered for discussion.

Many sequencing problems are combinations of these basic sequences and call for the use of several techniques when optimal solutions are sought. In Chapter 1 , a systematic method of constructing all of the feasible linear sequences which satisfy required precedence relationships has been developed. In Chapter 2 , several ways for analyzing and evaluating all possible alternative sequences according to certain criteria have been proposed. In Chapter 3, a systematic method of constructing all of the feasible compound sequences which are composed of feasible subsets of operations has been developed. Chapter 4 has dealt with the problem of determining an optimal compound sequence which is composed of subsets of operations to minimize the sum of subset values associated with them with precedence restrictions, and the line balancing problem. An effective algorithm to each of the problems has been developed.

In Part II , the insights into the role of inventory banks of production lines have been discussed. Management is responsible not only for answering how many stages to employ in the line, and in what order to place the stages, but also for analyzing how much interstage storage capacity to provide, and how to allocate the storage capacity among the stages. In Chapter 5, a theoretical study on the role of buffer stocks in production lines has been given.

In Chapter 6, the problems above for two stage lines have been answered. In Chapter 7, a very useful computer sumulation model to investigate the behavior of production lines has been developed. In Chapter 8, the above problems for multi-stage lines have been answered, based on the simulation model introduced in Chapter 7. From this research, inventory banks have been shown to be useful in some cases in improving the line efficiency of a production system since they reduce the effect of breakdowns at the stations. The general conclusions are :

(1) If there is no buffer, the line efficiency decreases markedly as the number of stages increases. This tendency becomes even more noticeable as the breakdown rates increase. The efficiency decrease rate per stage decreases as the number of stages decreases. The line efficiency improvement by providing buffer storage can be brought about effectively as the number of stages increases and as the breakdown rates get higher. As far as the line efficiency curves are concerned, there are three kinds of line efficiency curves E-1, E-2, and E-3. In the case of E-1, it is hopeless to improve the line efficiency by providing inventory banks. In the case of E-3, the line efficiency improves remarkably at some initial increase of buffer capacity, but thereafter the improvements get progressively much smaller. In the case of E-2, the question of how much buffer capacity should be prepared arises and careful cost analysis is required. As the number of stages increases, the line tends to be of type E-2 rather than type E-1, or type E-3 rather than type E-2. Therefore, as the number of stages increases, it becomes

necessary to provide buffer facility if the breakdown rates of the stages are high. At each increase of buffer capacit the line efficiency improvement gets progressively smaller, but as the number of stages increases, the buffer capacity which brings about the effects of installing buffer storage increases.

(2) There are three kinds of mean buffer stock curves B-1, B-2, and B-3. Type B-1 represents the case in which the mean buffer stocks increase linearly as buffer capacity increases ; Type B-2, the case in which the mean buffer stock curve is almost linear at first and then turns to be smooth concave and finally tends to converge to a certain value ; Type B-3, the case in which at each increase of buffer capacity the increase rate of the mean buffer stocks gets progressively smaller, and the mean buffer stock curve tends to converge to a certain value.

(3) Installing buffer storage is most effective when the breakdown rates are high and the mean repair time is short.

(4) Variation of breakdown rates affects the line efficiency more strongly than that of repair rates.

(5) If the difference of breakdown rates of a two stage line is small, the line efficiency takes the intermediate value between the line efficiency gained in the case where both the stages have the lower breakdown rates and that gained in

the case where both the stages have the higher breakdown rates. The variation of breakdown rates reduces the effects of installing inventory banks, while that of repair rates does not. Therefore, the stages should be designed to have approximately the same breakdown rate.

(6) Allocation ratio of the total buffer capacity to the buffers varies, depending upon the total buffer capacity. This allocation ratio tends to converge to a certain value as the total buffer capacity increases. When the stages have identical breakdown rates and identical repair rates, it is reasonable to allocate more buffer capacity to the preceding buffers.

(7) Unless technical considerations primarily determine the division of the line, the bad stage should follow other better stages.

The results would provide some guide to the role of inventory banks of production lines.

ACKNOWLEDGMENTS

The author wishes to express his gratitude for the guidance and encouragement received from Dr. K. Okamura.

BIBLIOGRAPHY

PART I

INTRODUCTION

1) Muth, J, F. and G.L. Thompson, *Industrial Scheduling*, Prentice-Hall (1963).

2) Pounds, W.F., *The Scheduling Environment*, Chapter 1 of the above 1).

3) Conway, R.W., W.L.Maxwell, and L.W.Hiller, Theory of Scheduling, Addison-Wesley (1967).

CHAPTER 1

1) Prenting, T.O. and R.M.Battaglin, "The Precedence Diagram : A Tool for Analysis in Assembly Line Balancing, Jound of Industrial Engineering, Vol. XV (1964), pp. 208-213.

2) Klein, M., "On Assembly Line Balancing," Oprn. Res., Vol. 11 (1963), pp. 274-281.

3) Ignall, E.T., "A Review of Assembly Line Balancing," Journal of Industrial Engineering, Vol. 16 (1965), pp. 244-254.

CHAPTER 2

1) McNaughton, R., "Scheduling with Deadline and Loss Functions, "Management Sci., Vol. 6 (1959), pp. 1.

2) Smith, W.E., "Various Optimizers for Single-Stage Production, " Nav. Res. Log. Quart., Vol. 3 (1956), pp. 182.

3) Gapp, W., et al., "Sequencing Operations To Minimize In-process Inventory Costs," Management Sci., Vol. 11 (1965), pp. 476.

4) Bowden, E.K., "Priority Assignment in a Network of Computers," IEEE, C-18, Vol. 11 (1969), pp. 1021.

5) Bellmore, M. and G.L. Nemhauser, "The Traveling Salesman Problem : A Survey," Oprn. Res., Vol. 16 (1968), pp. 538-558.

6) Bellman, R., "Dynamic Programming Treatment of the Traveling Salesman Problem," J. of ACM, Vol. 9 (1962), pp. 61-63.

7) Dantzig, G.B., D.R.Fulkerson, and S.M.Johnson, "Solution of a Large Scale Traveling Salesman Problem," Oprn. Res., Vol. 2 (1954), pp. 393-410.

8) Gilmore, P.C., and R.E.Gomory, "Sequencing a One-State Variable Machine : A Solvable Case of the Traveling Salesman Problem," Oprn. Res., Vol. 12 (1964), pp. 655-679.

9) Hardgrave, W.W. and G.L.Nemhauser, "On the Relation Between the Traveling Salesman and the Longest Path Problem," Oprn. Res., Vol. 10 (1962), pp. 647-657.

10) Held, M. and R.M.Karp, "A Dynamic Programming Approach to Sequencing Problems," SIAM, Vol. 10 (1962), pp. 196-210.

11) Held, M. and R.M.Karp, "The Traveling Salesman Problem and Minimum Spanning Trees : Part II ," Mathematical Programming, Vol. 1, 1 (1971).

12) Karg, L.L. and G.L.Thompson, "A Heuristic Approach to Solving Traveling Salesman Problems," Management Sci., Vol. 10 (1964), pp. 225-248.

13) Lin, S., "Computer Solution of the Traveling Salesman Problem," Bell System Tech. J. 44 (1965), pp.2245-2269.

14) Little, J.D.C., K.G.Murty, D.W.Sweeney, and C.Karel, "An Algorithm for the Traveling Salesman Problem," Oprn. Res. Vol. 11 (1963), pp. 979-989.

15) Miller, C.E., A.W. Tucker, and R.A. Zemlin, "Integer Programming Formulations and Traveling Salesman Problems," J. of ACM, Vol. 7 (1960), pp. 326-329.

16) Roberts, S.M. and B. Flores, "An Engineering Approach to the Traveling Salesman Problem," Management Sci., Vol. 13 (1966), pp. 269-288.

17) Rothkopf, M. "The Traveling Salesman Problem : On the Reduction of Certain Large Problems to Smaller Ones," Oprn. Res., Vol. 14 (1966), pp. 532-533.

18) Efroymson, M.A. and T.L.Ray, "A Branch-Bound Algorithm for Plant Location," Oprn. Res., Vol. 14 (1966), pp. 361-368.

19) Gavett, J.W. and N.V.Plyter, "The Optimal Assignments of Facilities to Locations by Branch and Bound," Oprn. Res., Vol. 14 (1966), pp. 210-232.

20) Geoffrion, A.M., "Integer Programming by Implicit Enumeration and Balas' Method," SIAM Review, Vol. 9 (1967), pp. 178-190.

21) Greenberg, H.H., "A Branch-Bound Solution to the General Scheduling Problem," Oprn. Res., Vol. 15 (1967), pp. 353-361.

22) Ignall, E. and L.E.Schrage, "Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems," Oprn. Res., Vol. 13 (1965), pp. 400-412.

23) Kolesar, P.J., "A Branch and Bound Algorithm for the Knapsack Problem," Management Sci., Vol. 13 (1967), pp. 723-735.

24) Lawler, E.L., "The Quadratic Assignment Problem," Management Sci., Vol. 9 (1963), pp. 586-599.

25) Lawler, E.L. and D.E.Wood, "Branch-and Bound Methods : A Survey," Oprn. Res., Vol. 14 (1966), pp. 699-719.

26) Lawler, E.L., "On Scheduling Problems with Deferral Costs," Management Sci., Vol. 11 (1964), pp. 280-288.

CHAPTER 4

1) Jackson, J.R., "A Computing Procedure for a Line Balincing Problem," Management Sci., Vol. 2 (1956), pp. 261-271.

2) Helgeson, W.B. and D.P.Birnie, "Assembly Line Balancing Using the Ranked Positional Weight Technique," Journal of Industrial Engineering, Vol. XII (1961), pp.394-398.

3) Kilbridge, D.M. and L.Wester, "The Balance Delay Problem," Management Sci., Vol. 8 (1961), pp. 69-84.

4) Bowman, E.H., "Assembly Line Balancing by Linear Programming," Oprn. Res., Vol. 8 (1960), pp. 385-389.

5) Gutjahr, A.L. and G.L. Nemhauser, "An Algorithm for the Line Balancing Problem," Management Sci., Vol. 11 (1964), pp. 308-315.

6) Held, M., R.M.Karp and R.Shareshian, "Assembly-Line Balancing-Dynamic Programming with Precedence Constraints," Oprn. Res., Vol. 11 (1963), pp. 442-459.

7) Arcus, A.L., "COMSOAL A Computer Method of Sequencing Operations for Assembly Lines," Int. J. Prod. Res., Vol. 4 (1966), pp. 259-277.

8) Freeman, D.R. and J.V. Jucker, "The Line Balancing Problem," Journal of Industrial Engineering, Vol. XIX (1967).

9) Hoffman, T.R., "Assembly Line Balancing with a Precedence Matrix," Management Sci., Vol. 9 (1963), pp. 551-562.

10) Kilbridge, D.M. and L.Wester, "A Review of Analytical Systems of Line Balancing," Oprn. Res., Vol. 10 (1962), pp. 626-638.

11) Kilbridge, D.M. and L.Wester, "A Heuristic Method of Assembly Line Balancing," Journal of Industrial Engineering, Vol. XII (1961), pp. 292-298.

12) Mansoor, E.M., "Assembly Line Balancing-An Improvement on the Ranked Positional Weitht Technique," Vol. XV. (1964), pp. 73-77.

13) Moodie, C.L. and H.H. Young, "A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work Element Times," Journal of Industrial Engineering, Vol. XVI (1965), pp. 23-29.

14) Ramsing, K. and R.Downing, "Assembly Line Balancing with Variable Element Times," Industrial Engineering, (1970)

15) Reiter, R., "On Assembly-Line Balancing Problems," Oprn. Res. Vol. 17 (1969), pp. 685-700.

16) Tonge, E.M., A Heuristic Program of Assembly Line Balancing, Prentice-Hall, (1961).

17) Tonge, E.M., "Assembly Line Balancing Using Probabilistic Combinations of Heuristics," Management Sci., Vol. 11 (1965), pp. 727-735.

18) Wester, L. and M.Kilbridge, "Heuristic Line Balancing : A Case," Journal of Industrial Engineering, Vol. XIII (1962), pp. 139-149.

PART II

CHAPTER 5

1) Hunt, G.C., "Sequential Arrays of Waiting Lines," Oprn. Res., Vol. 4 (1956), p.674.

2) Koenigsberg, E., "Production Lines and Internal Storage," Management Sci., Vol. 5 (1956), p. 410.

3) Morse, P.M., Queues, Inventories, and Maintenance, John Wiley and Sons, Inc., New York, (1958).

4) Elmaghraby, S.E., The Design of Production Systems, Chapter 5, Reinhold, (1966).

5) Buchan, J. and Koenigsberg,E., Scientific Inventory Management, Chapter 22, Prentice-Hall, (1963).

6) Patterson, R.L., "Markov Processes Occurring in the Theory of Traffic Flow Through an n-State Stockastic Flow System," Journal of Industrial Engineering, Vol. XV (1964), pp. 188-193.

7) Hillier, F.S. and R.W.Boling, "The Effect of Some Design Factors on the Efficiency of Production Lines with Variable Operation Times," Journal of Industrial Engineering, Vol XVII (1966).

8) Hillier, F.S., and R.W.Boling, "Finite Queues in Series with Exponential or Erlang Service Times-A Numerical Approach," Oprn. Res., Vol. 15 (1967).

9) Love, R.F., "A Two-Station Stochastic Inventory Model with Exact Methods of Computing Optimal Policies," Nav.Res. Log. Quart., Vol. 14 (1967).

10) Hatcher, J.M., "The Effect of Internal Storage on the Production Rate of a Series of Stages Having Exponential Service Times," AIIE Transactions, Vol. 1, No. 2 (1969), pp. 150-156.

11) Buzacott, J.A., "Automatic Transfer Lines with Buffer Stocks," Int. J. Prod. Res., Vol. 5 (1967), pp. 183-200.

12) Buzacott, J.A., "Prediction of the Efficiency of Production Systems without Internal Storage," Int. J. Prod. Res., Vol. 6 (1968), pp. 173-188.

13) Buzacott, J.A., "The Role of Inventory Banks in Flow-Line Production Systems," Int. J. Prod. Res., Vol. 9, (1971), pp. 425-436.

14) Knott, A.D., "The Efficiency of a Series of Work Stations-A Simple Formula," Int. J. Prod. Res., Vol. 8 (1970), pp. 109-119.

15) Kay, E., "Buffer Stocks in Automatic Transfer Lines," Int. J. Prod. Res., Vol. 10 (1972), pp. 155-165.

CHAPTER 7

1) Barten, K., "A Queueing Simulator for Determining Optimum Inventory Levels in a Sequential Process," Journal of Industrial Engineering, Vol. XIII, (1962), pp. 245-252.

2) Freeman, M.C., "The Effects of Breakdowns and Interstage Storage on Production Line Capacity," Journal of Industrial Engineering, Vol. 15 (1964), pp. 194

3) Young, H.H., "Optimization Models for Production Lines," Journal of Industrial Engineering, Vol. 18 (1967), pp. 70-78.

4) Anderson, D.R. and C.L.Moodie, "Optimal Buffer Storage Capacity in Production Line Systems," Int. J. Prod. Res., Vol. 7 (1969), pp. 233.